NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TRIAGE VISUALIZATION FOR DIGITAL MEDIA EXPLOITATION

by

Glenn Henderson

September 2013

Thesis Advisor:	Simson Garfinkel
Second Reader:	Rudolph Darken

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
25–9–2013	Master's Thesis	2010-09-12—2013-09-27

4. TITLE AND SUBTITLE

TRIAGE VISUALIZATION FOR DIGITAL MEDIA EXPLOITATION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Glenn Henderson

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

14. ABSTRACT

Digital forensic examiners are overwhelmed by case loads and data volumes and must prioritize their work. This thesis hypothesis that digital forensic examiners can employ triage visualizations to prioritize work loads. This thesis presents a simple one page visualization of disk activity for Windows FAT and NTFS filesystems. The visualization is constructed from filesystem meta data carved by the open source *bulk_extractor* digital forensics application. The visualization does not require further examination or reconstruction of file system metadata. The visualization is able to detect minor obfuscation or modification and overwriting of file system timestamps.

15. SUBJECT TERMS

digital forensics, visualization, triage

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	79	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TRIAGE VISUALIZATION FOR DIGITAL MEDIA EXPLOITATION

Glenn Henderson
Civilian, Vista Research Inc.
B.S., James Madison University, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Glenn Henderson

Approved by: Simson Garfinkel
Thesis Advisor

Rudolph Darken
Second Reader

Peter J Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Digital forensic examiners are overwhelmed by case loads and data volumes and must prioritize their work. This thesis hypothesis that digital forensic examiners can employ triage visualizations to prioritize work loads. This thesis presents a simple one page visualization of disk activity for Windows FAT and NTFS filesystems. The visualization is constructed from filesystem meta data carved by the open source *bulk_extractor* digital forensics application. The visualization does not require further examination or reconstruction of file system metadata. The visualization is able to detect minor obfuscation or modification and overwriting of file system timestamps.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

CDF	Cumulative Distribution Function
DOI	Denial of Information
FBI	Federal Bureau of Investigation
NPS	Naval Postgraduate School
MAC	Modify, Access, Create
MFT	Master File Table
PDF	Portable Document Format
RCFL	Regional Computer Forensics Laboratory
USG	United States Government
US-CERT	United States Computer Emergency Readiness Team

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

I would like to express my gratitude to my advisor Dr. Simson Garfinkel for his time and energy in helping me complete this thesis. I would like to thank my second reader Dr. Rudolph Darken for his comments and suggestions on my thesis. I would also like to thank Mike Shick for his work on the tcpflow timeline tool.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

This section describes the motivation and focus for this research and gives a brief overview of the thesis.

1.1 Motivation

The current models applied to the digital forensics field are not keeping pace with the volume of digital media bestowed upon digital forensic examiners. The growth in digital media size has exponentially increased the amount of processing required to perform basic forensic analysis. As it is unlikely that there will be a dramatic increase in the hiring of trained digital forensic examiners, new approaches to analyze digital material should be developed. Methods for triaging media for analysis need to be developed to handle the expansion of media size and the allocation of processing resources. Visualizations allow for faster identification of trends and patterns and can be employed to triage media in an effective and efficient manner. Digital forensic examiners can use triage visualizations to process incoming media and better organize their cases.

1.2 Research Focus

This research is focused on providing a visualization for digital forensic analysts to efficiently triage media. This visualization was developed under the following guidelines.

Simple The visualization must be simple in layout and design
Efficient The visualization must be generated efficiently
Truthful The visualization must represent the data truthfully without bias or skew

To triage digital media, analysts require information regarding the contents of the media. Once media has been imaged, automated tools can analyze and generate reports which allow an analyst to make decisions regarding which media needs to be examined further. The visualization in this research is intended to reveal important details to assist digital forensic analysts with media triage.

1.3 Thesis Layout

The second chapter of this thesis describes the previous work established in the field of digital forensics, visualization and triage analysis for digital forensic media. Chapter 3 describes the

triage process and the use of *bulk_extractor* on FAT and NTFS MFT file system structures as well as visualization techniques employed in this thesis. Chapter 4 describes the implementation of the visualization and metrics used to judge performance of the visualization. Chapter 5 presents a comparison between the visualization in this work and the visualization provided by a commercial tool. The final chapter of this thesis describes future work that can extend this visualization for greater use in the digital forensics field.

CHAPTER 2:
Literature Review

This chapter describes previous work in the digital forensics field and highlights those works involved in visualizations for digital forensics.

2.1 Digital Forensics

US-CERT defines computer forensics as "the discipline that combines elements of law and computer science to collect and analyze data from computer systems, networks, wireless communications, and storage devices in a way that is admissible as evidence in a court of law." [1] Recently the term computer forensics has been renamed to "digital forensics" to reflect the fact that devices other than computer are examined. One of the largest issues in digital forensics is the overwhelming amount of data that an analyst must process. In US Government fiscal year 2011 the FBI Regional Computer Forensics Laboratory processed 4,263 terabytes of data in the pursuit of their duties [2]. The proliferation of digital media has led to an explosion in digital forensic case loads. Given the overwhelming amount of data, a triage system must be put in place to allow investigators to prioritize analysis of incoming media.

Digital forensics relies on the scientific method to ensure accuracy and reproducibility, allowing digital evidence to be admitted in a courtroom. A general model for judicial or corporate investigations was proposed by Casey and includes 12 steps from "Incident alerts or accusation" to "Persuasion and testimony" [3]. The digital forensic investigation model introduced by Casey provides abstract steps that an investigator follows while examining digital evidence [3]. The investigative process model includes these steps:

1. Incident alerts or accusation
2. Assessment of worth
3. Incident/Crime scene protocols
4. Identification or seizure
5. Preservation
6. Recovery
7. Harvesting
8. Reduction
9. Organization and search

10. Analysis
11. Reporting
12. Persuasion and testimony

An investigation for which a judicial hearing may be necessary requires a methodical process that any investigator can are reproduce. The abstract steps allow for a rigorous scientific approach and accounting of evidence which can be applied to criminal, corporate or military investigations.

2.2 Visualization

The visualization field is concerned with displaying information to a viewer. The information is encoded in the visualization by mechanisms such as color, size, placement, etc. which are decoded by a viewer. The human eye and brain are able to perceive visual information in parallel, allowing for expedited processing of graphs and charts over plain text [4]. The viewer has to understand the encoding technique used by the author to understand the data presented in the visualization. To correctly apply encoding techniques, one requires analysis of the information being represented to find the best format to portray the data in a truthful and meaningful manner.

There are two primary goals of visualization that an author must take into account when designing a visualization. One goal requires minimal loss of significant information during the encoding process. The author must strive to include all necessary information to the viewer. While the degree of acceptable loss is an open research question, the visualization should attempt to preserve any data that do not conform to the overall message. Outliers should be presented so that the viewer is aware of any data point not conforming to the overall pattern of a data set. The second goal of visualization pertains to not misleading a viewer during the decoding of the information. These two goals complement each other in that the visualization must be truthful in representing the data.

The goal for the visualization designed in this research is to reduce what Tufte calls the ”Lie Factor” [5] as much as possible. The “Lie Factor” is a measure of how much skew is introduced when generating a visualization from data (encoding) and during the decoding by a viewer. The “Lie Factor” can be represented as the size of effect shown in a graphic divided by the size of the effect in the data it represents. A common example of misrepresentation is the use of deceptive scales or the introduction of a third dimension that does not scale to represent the data. The scaling of circles by radius as opposed to area is a common example of deceptive data rep-

resentation cited in the literature. The "Lie Factor" is particularly important in digital forensics visualization because of the large amount of data typically under investigation. A misrepresentation of digital forensic data can skew an investigation costing critical time to apprehend an offender or resources in investigating ineffective leads.

2.3 Computerized Visualization

The practical application of computers to visualization techniques allows for efficient generation of visually appealing graphics for variable size data sets. Visually appealing graphics help people understand high entropy information better [6] [7]. Computers allow for efficient and automated generation of multiple graphic images for presentation to the viewer. The visualization presented in this research relies on computer algorithms to generate a graphic that is visually appealing and understandable to a diverse audience.

The use of computer visualizations in digital forensics allow for faster processing of digital evidence and standardized representations of information. Computer visualizations can efficiently represent large data sets typical of digital forensic caseloads. Computer visualizations can be generated from automated media analysis tools without investigator input which accelerates the investigation process. As the standard hard drive size of commodity computers grows, forensic investigators need utilities that can generate high entropy representations of all the information contained on a media device.

2.4 Digital Forensics Visualization

Digital forensics visualization is a relatively new field combining the techniques from the visualization field with the data of digital forensics. As caseloads increase, digital forensic analyst require tools to efficiently process large amounts of information. Digital forensic visualization can be employed in the "Reduction" through the "Persuasion and testimony" steps of the digital forensic process described by Casey. Graphical representations of trends using such meta data as timestamps can help in scoping an investigation and identifying the "patterns of life" in digital media. The "Organization and search" step can be assisted by visualization of the groups and tags used to place evidence into meaningful units. Visual representation can help investigators find and identify meaningful data within media and records to help organize and prioritize searches. The "Analysis" step can benefit from visualizations that help in finding trends and links between data discovered during the "Reduction" step. Visualizations can assist in the "Reporting" and "Persuasion and testimony" steps, particularly when they are concerned

with trends and discrete data sets occurring in digital evidence. Visualizations are also useful in the courtroom to help explain findings to a non-technical jury. This last use is only relevant if the visualizations are simple enough to be decoded and understood by a large population or can be explained sufficiently by a technical expert.

2.4.1 Digital Forensic Visualization Techniques

This section describes a variety of existing digital forensic visualization techniques that have been presented in the scientific community.

Timelines

Timeline analysis is the use of time-based events in digital media to explain when events occurred in "real time". Timelines are commonly composed of timestamps from files on a filesystem. Modern operating systems associate multiple timestamps with each file in a filesystem and can be utilized to track one or many users on an information system [8]. The timeline has many representations, but typically contains a linear time axis with event occurrence or frequency marked according to their time. Timelines can delineate when events of interest occurred in the time scope of the investigation. Timelines allow for easier correlation to other evidence that is linked to a case in time.

The Digital Forensics Research Workshop (DFRWS) holds an annual forensic challenge open to the public. Each challenge is marked by a scenario in which digital evidence is acquired in the due course of an investigation. In 2011 the DFRWS held a challenge for the analysis of an two Android devices, one of which contained information regarding a mysterious death while the other indicated a case of intellectual property theft [9]. The submissions were required to contain a reconstructed timeline of relevant events on each device. The winning submission by Fox-IT included a novel visual reconstruction of a timeline to assist in reconstructing the events of the case seen in Figure 2.1 and Figure 2.2.

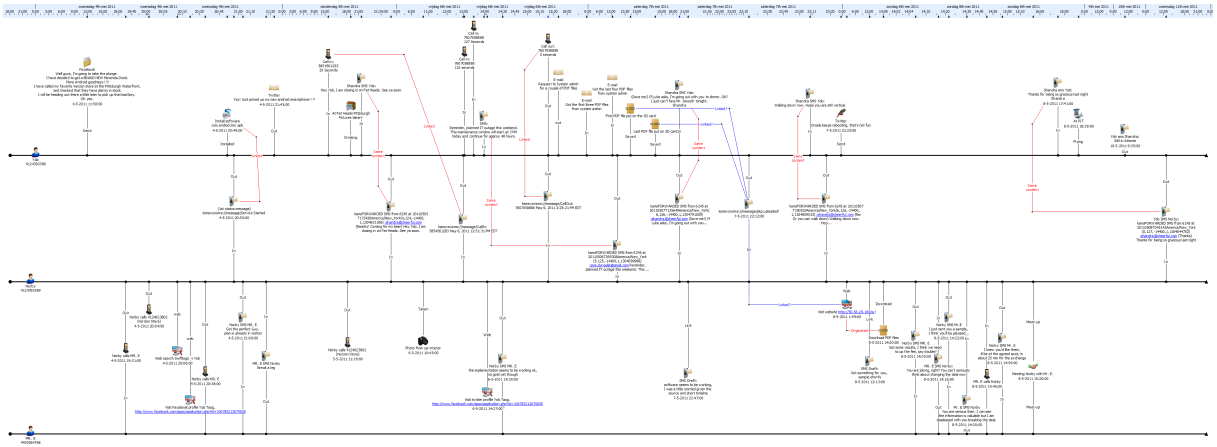


Figure 2.1: The Fox-IT visual reconstruction generated for the DFRWS 2011 Challenge. From [9].

Hagreaves and Patterson demonstrated how timelines can be employed in the reconstruction of high-level events from low level timestamps at the DFRWS 2012 Conference [10]. Their tool, Timeflow, allows for descriptions of low-level event filters that produce high-level events which are used to generate a timeline. Once high-level reconstruction is complete the generation of a graphic with multiple views showing a timeline is presented for analysis.

Another approach to timeline visualization was introduced by Olsson and Boldt in the development of the Cyber Forensics Time Lab (CFTL) [8]. The utility allows for generating a timeline of evidence by scraping files for timestamps. CFTL was shown to be faster for solving a fictional case than the commercial tool Forensic Tool Kit (FTK).

Treemap

Treemaps are another method for reducing a large amount of numerical data to a single image. Treemaps (Figure 2.3) can be used to visualize blocks of related information. Each block in the tree-map represents a specific set of data that is related by some attribute. The size of each block is calculated from the frequency of the attribute being represented. Treemaps can be used for basic filtering on parameters such as file size and hierarchical file system position [11].

Self Organizing Maps

Self organizing maps (SOMs) are a biologically inspired approach to finding patterns in data sets [14]. The algorithm for SOMs was conceived in 1981 by Teuvo Kohonen for use in artificial intelligence systems. The self organizing map can be presented as a visualization tool to help an investigator find patterns within digital evidence for further analysis [15]. SOMs are built on a neural network model that maps high-dimensional data onto low-dimensional space, such as two-dimensional graphics. SOMs have been applied to network traffic for analysis of network attack and anomalous network behaviour with success [16].

Network Visualization

Visualization has been studied in networking environments as a way to understand complex interconnected networks. The use of computer graphics to ascertain high-level details of interconnected networks allow for tracing flows of network data across network boundaries. Graphic techniques commonly deployed in the network visualization field include cyclic, tree and force directed graphs for representing nodes within a networked environment.

Network visualization utilities allow for situational awareness for computer networks in identifying and responding to threats. Previous work includes NVisionIP [17], a network visualization

tool that processes Argus NetFlow [18] data. NVisionIP allows an analyst to capture a single view of a class B network. The visualization allows for drilling down into a collection of hosts as well as a single host using a variety of bar graphs to represent network information.

Greg Conti's PhD dissertation describes how denial of information (DOI) attacks can be used to thwart the human capability to decipher fact from fiction in digital information [19]. The computer security visualizations described and implemented in Conti's thesis show a reduction in the amount of superfluous information presented to a user to make an accurate threat assessment.

The application tcpflow [20] contains a summary page visualization upon which this work is based. The tcpflow network visualization produces a summary graphic which includes a packet timeline and a sorting of packets into types along a histogram as well as secondary views of top source and destination addresses and ports. This visualization allows for efficient analysis of the summary of a network trace into high level components.

2.4.2 Digital Forensics Triage

The digital forensics triage process occurs upon initial acquisition and imaging of any media as evidence. Investigations may require immediate feedback from digital evidence in the case of abductions or a threat to life. Triage models for law enforcement have been used successfully to obtain leads on-site from digital evidence that can be used in prosecution in the courtroom [21]. The triage of digital devices such as cell phones with unknown storage formats can be accomplished using data-driven programming techniques with high probability of success [22]. The triage use of *bulk_extractor* for bulk media analysis, extraction of features from media instead of files, allows an investigator to obtain information early in the acquisition process [23].

Triage Visualization

Triage visualizations are comprised of summary information about media being investigated and are used to prioritize media for further investigation. Reports generated for triage analysis should allow for efficient removal of unnecessary media. The summary reports can be used to triage media for further analysis [24].

Current tool sets require trained professionals and present summary results poorly for triage usage. New architectures are needed to build visualization systems that enhance forensic investigation. The thesis by Farrell highlights well defined reporting metrics and includes a sample triage visualization for use in media exploitation [25]. Data presented in familiar and standard-

ized formats can speed analysis and allow for easier access to context relevant information. The integration of statistics and machine learning with visualization techniques can help an investigator focus on outliers and patterns [26].

2.4.3 Commercial Tools

The Forensic Toolkit is a commercial utility for digital forensic investigations [27]. FTK processing is based upon a law enforcement model where media is handled on a per case basis. FTK provides visualization features as an add-on capability. FTK visualizations provide graphical views of meta-data relating to evidence, commonly files extracted from media. The visualization examined in this thesis is the timeline analysis of file timestamps extracted from FTK's data ingress processing.

2.4.4 Open-Source Tools

Autopsy is an open-source graphical front-end to The Sleuth Kit which is a collection of libraries and command line utilities for examining disk images [28]. As of version 3.0.5, Autopsy has included a Timeline visualization feature (Figure 2.5) that will display a time histogram of the modify timestamps on file objects parsed from a disk image. The Sleuth Kit analyzes the entire file system metadata from the top level directory to construct the timestamp listing used in the timeline visualization. The Autopsy timeline visualization can be scaled to finer time ranges by selecting a histogram bar, effectively zooming in on a smaller time range.

2.4.5 Digital Forensics Tool Validation

Validation of digital forensics tools against real evidence or secondary market acquired hard drives is difficult because the actual activity, known as ground truth, on the drive is not known. Research into generating images for which ground truth is known, but randomness can be introduced, is currently on-going [29]. Until images can be generated for research that emulate real life digital forensic investigation scenarios, validation of tools, such as the one presented in this thesis, is made with images created manually.

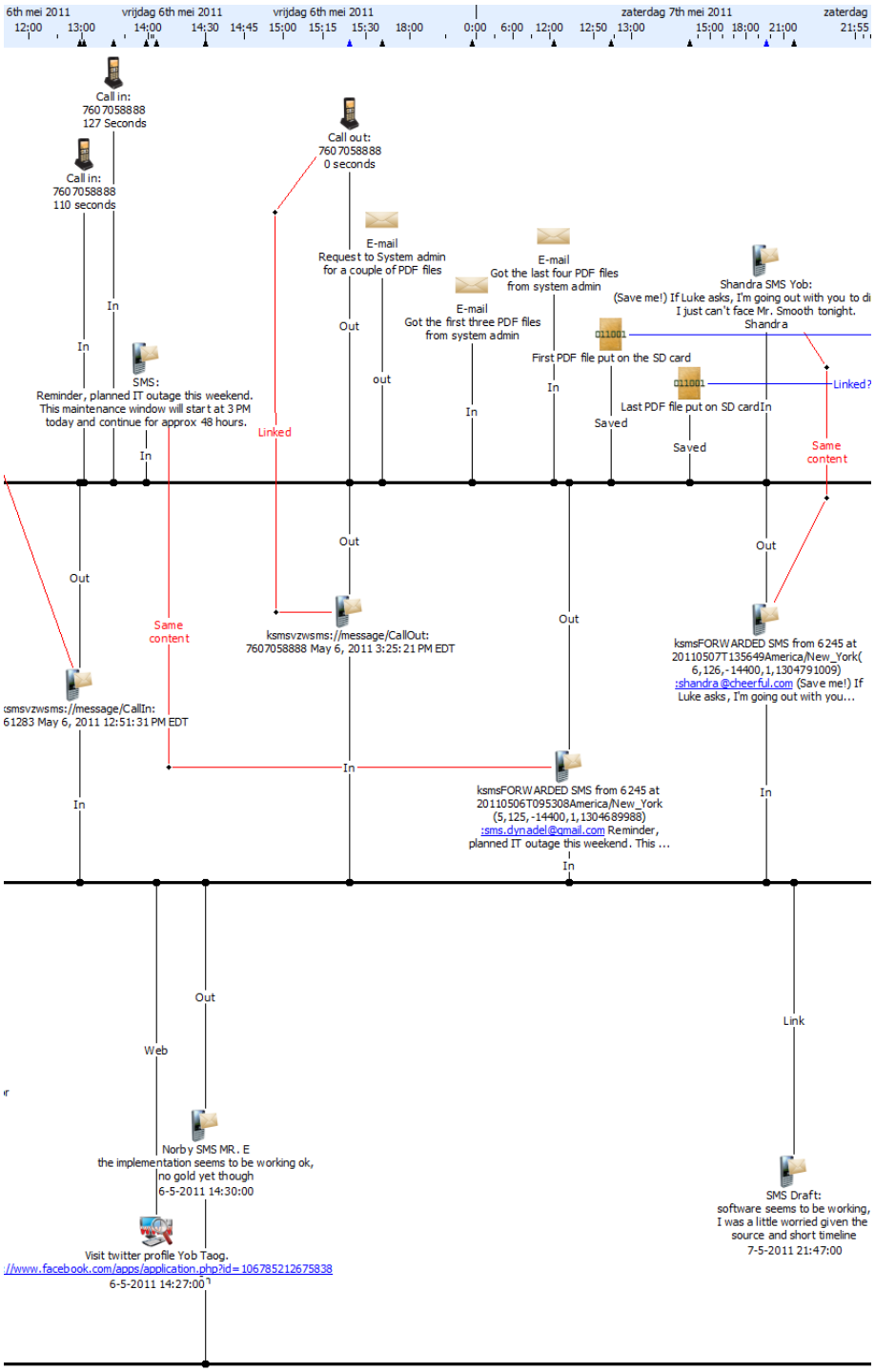


Figure 2.2: An enlarged excerpt of the Fox-IT visual reconstruction for the DFRWS 2011 Challenge. From [9].

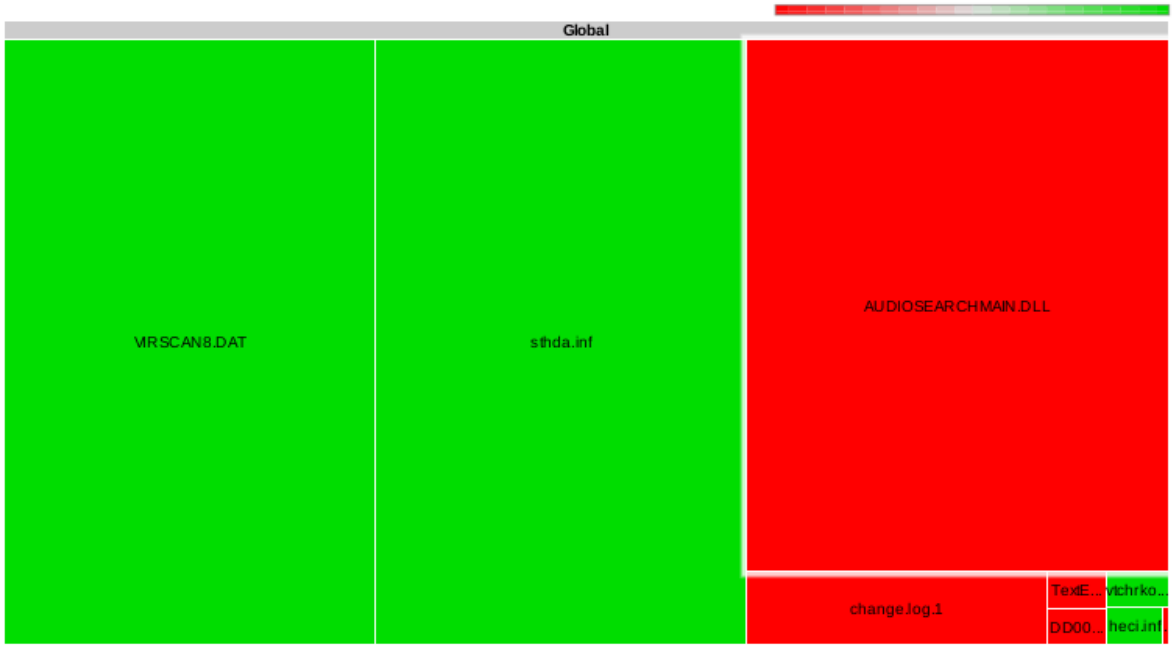


Figure 2.3: A treemap generated using Google charts [12] displaying file sizes (object size) and allocation (color) from the M57-Jean scenario. After [13].

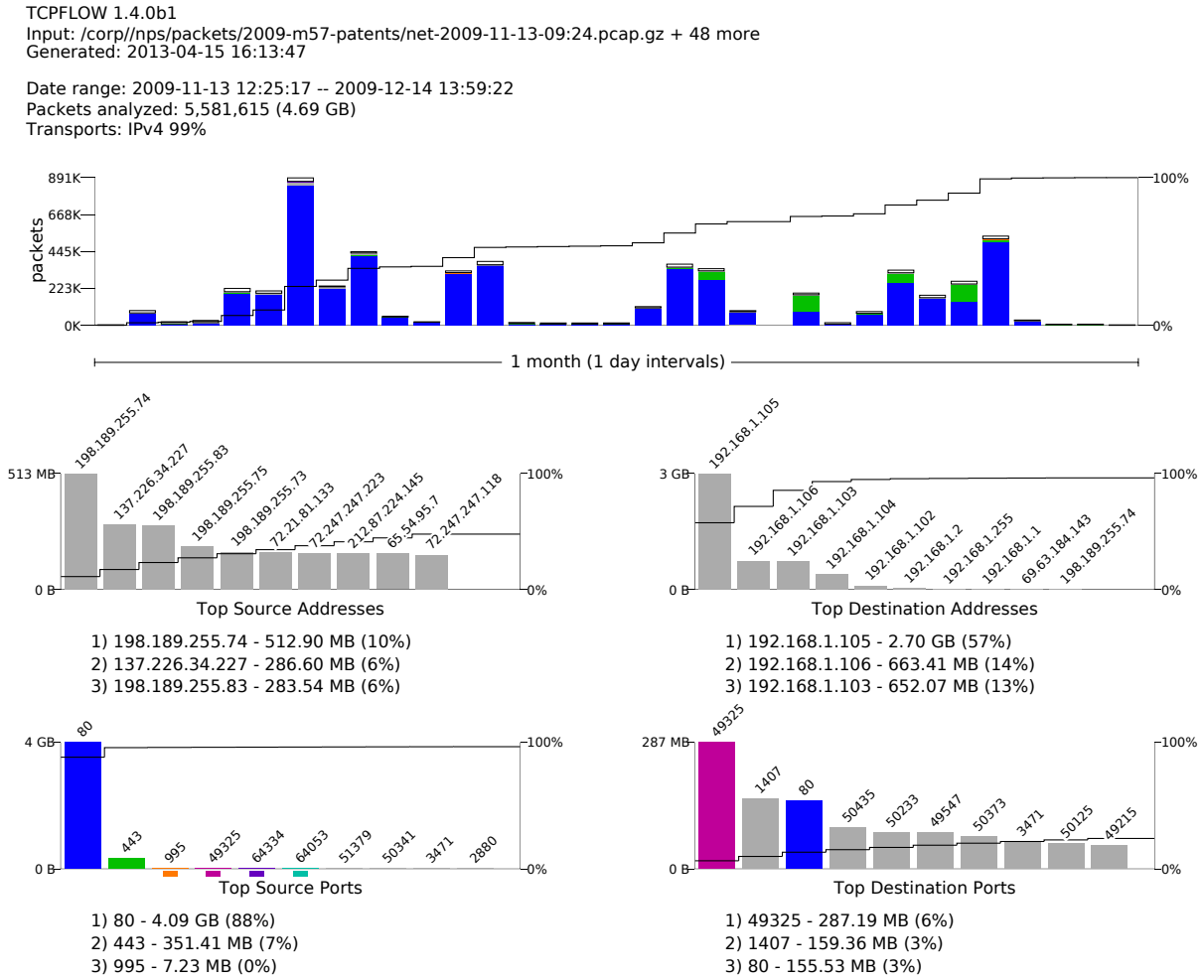


Figure 2.4: The tcpflow summary representation of the M57-Jean scenario at Digital Corpora [13]. From [20]

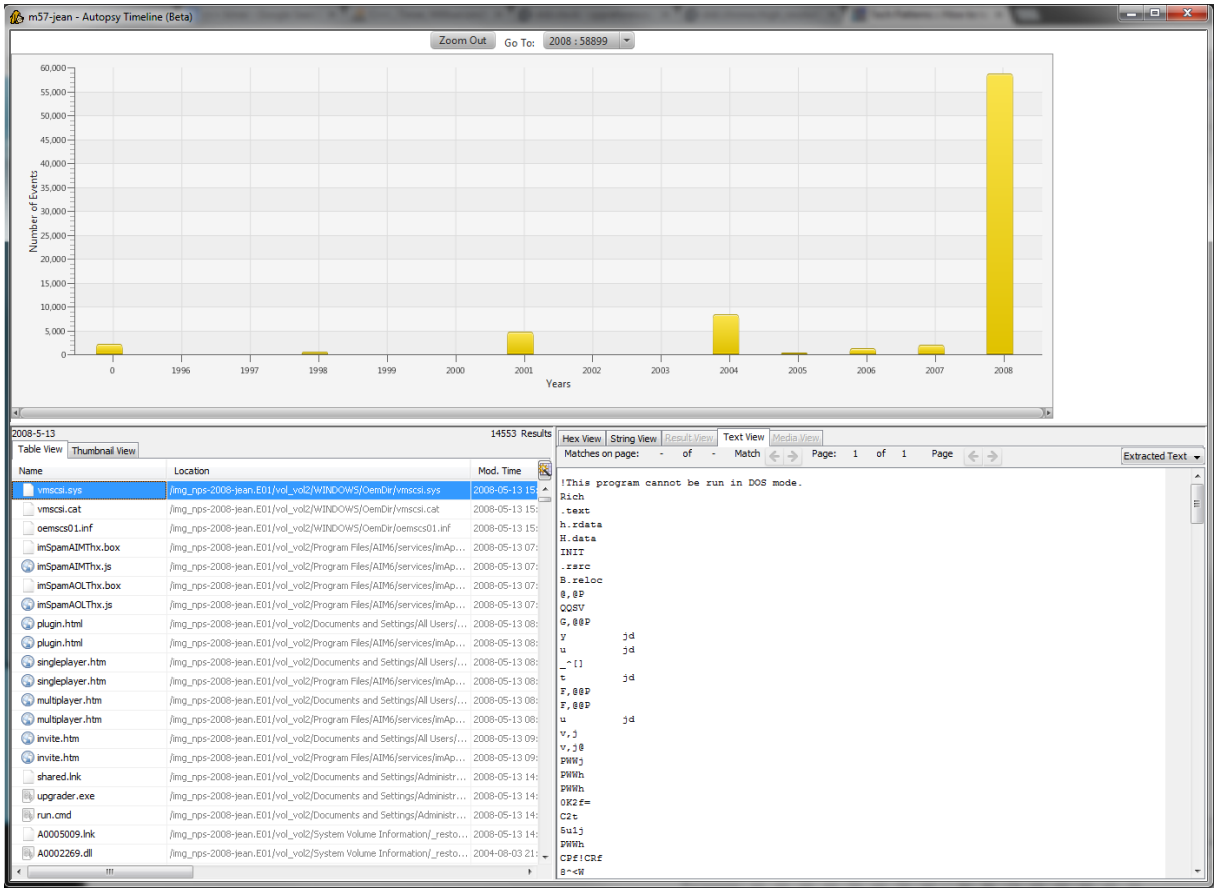


Figure 2.5: The Autopsy timeline display generated from the M57-Jean scenario posted on digitalcorpora. After [28].

CHAPTER 3:
Digital Forensics Triage Visualizations

This section describes the digital forensics triage process including tools and concepts used in generating the disk activity timeline.

3.1 Triage

The triage process in digital forensics refers to the evaluation and prioritization of media based upon the media contents. Media contents include filesystem information, file content, timestamps and any information that may be relevant to a digital forensic investigation. The triage process begins after a forensic copy of the media has been obtained and logged as evidence. The growth of media storage capacity makes the initial analysis time consuming if one is attempting an exhaustive parsing of all of the files on a file system. An investigation may require immediate information regarding media, in which case a full analysis is not possible. The results from a triage analysis are used to assist the forensic investigator in making a determination of whether the digital media may contain any evidence of value. The triage results assist a forensic investigator in prioritizing media for more rigorous investigation.

Triage visualizations can be used by digital forensics investigators to find important and relevant patterns within the media that they analyze. Visualization can allow a forensic investigator to ascertain the period of activity on the media and quickly determine if relevant events may have happened in a timeframe of interest. The primary goal of the triage visualization is to allow the forensic investigator to rapidly analyze the outline of activity and events on the media to determine if it may contain relevant data.

Effective triage visualizations are be simple to understand, efficiently generated and contain only factual information about the media under examination. The simplicity of a visualization is not an indication of the amount of information presented to the viewer, but an indication of how quickly an investigator can ascertain the salient facts about the media under investigation. Facets of a simple visualization include correct use of color palettes to highlight important details as well as accurate and consistent scaling to ensure no data is skewed for the viewer. The triage process implies that the visualization is generated in a timely manner. No metric exists for an acceptable timeline that is dependent on media size, but the visualization cannot take longer to generate than examining the media using traditional tools. The visualization is

factual so that a forensic examiner can make a provable determination of the priority of the media for examination. Graphics used in the visualization represent the data without omission and annotate any variation caused by rendering or scaling.

3.2 Requirements

An effective digital forensic visualization is simple to understand, but contains enough information to prioritize media for further investigation. This requirement is taken into account for each metric that is added to a digital forensic visualization. If a metric does not add significant value to the decision of priority than it is considered extraneous and left out. Disk activity timelines represent frequency of activity over time and pertain to many types of media. Features other than timelines are more difficult to apply to all media types. Digital forensic visualizations must adapt to the media supplied as input. Each feature has to be judged based on the practical application to an investigation and the usefulness in making a triage determination.

A digital forensic visualization must scale according to the input source. Media sources come from a variety of hardware from single user hard drives and mobile phones to large networked computer systems. Depending upon the media source, a visualization must be able to denote each input as well as display all relevant summary data. The current implementation presented in this thesis is directed to a single drive, single user, case scenario. Future work may extend the capabilities of the visualization to cover a larger number of inputs.

3.3 bulk_extractor

bulk_extractor is a bulk media analysis program [23]. Bulk media analysis allows for processing of disk images from start to finish without regard to file system meta data. Data from the input media are not processed as files independently, instead the application divides pieces of the disk into chunks and processes each chunk. Bulk media analysis has the advantage of working with any media format. Bulk media analysis is suited to the triage application as it is efficient, automated and highly parallelizable. While the analysis does not include the complete details of the filesystem it may be able to extract important features to assist in prioritizing media.

bulk_extractor processes media as chunks, commonly referred to as pages, and allows for opportunistic decompression of compressed chunks as well as parallel processing of chunks. As each chunk is decompressed from the media stream it is processed by feature scanners as a new chunk to be re-examined (see Figure 3.1). The recursive re-examination allows for feature extraction of compressed data in an efficient manner. *bulk_extractor* and bulk media analysis in

general is easily parallelized for fast computation of pages. The speed of the processing allows for fast automated triage analysis of media. The automated nature of *bulk_extractor* allows for efficient triage analysis of media without human interaction and extraction of features of interest to forensic investigators.

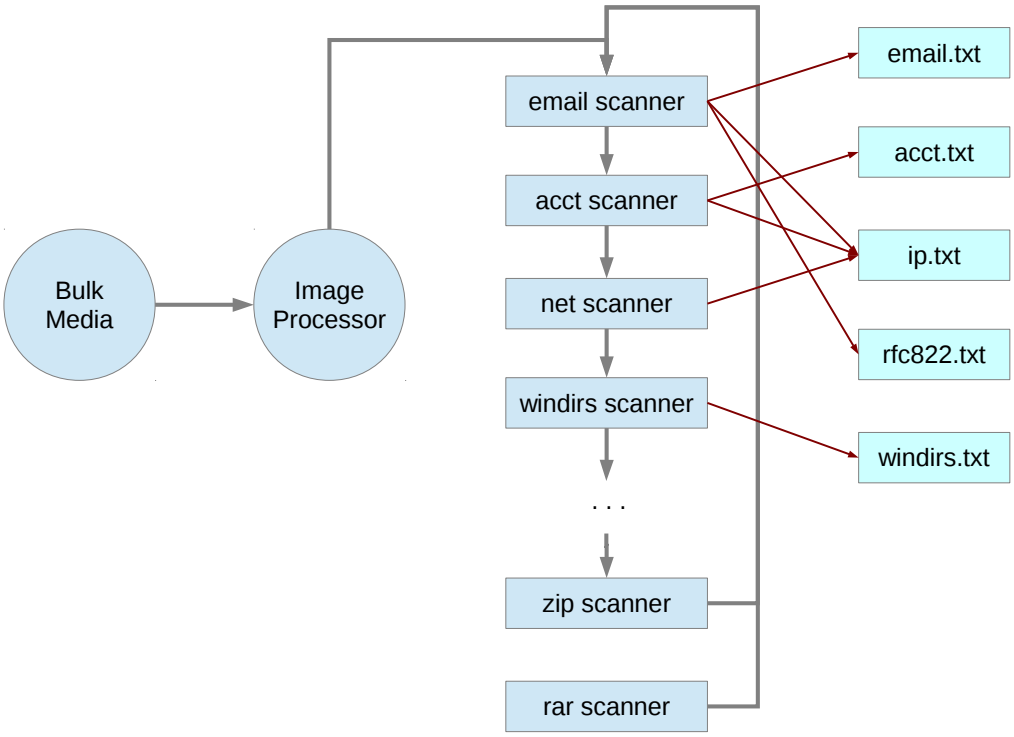


Figure 3.1: An abstract view of *bulk_extractor* processing chain and feature file population.

3.3.1 Feature Extraction

The feature extraction component of *bulk_extractor* generates plain text tab-delimited files for each scanner that is enabled. This thesis uses the *windirs* feature extraction file which contains information that results from parsing FAT and NTFS directory entries.

Currently *bulk_extractor* is able to recover file system directory information from FAT12, FAT16, FAT32, and NTFS file system meta data. The *windirs* feature file is composed of a five line header describing the format and source of the data and then a line for each feature item extracted. Each feature line includes the location of the item in the media image as well as a name and an XML string containing relevant attributes extracted for each feature. The sample

in Figure 3.2 is from the *windirs* file generated from an example forensic scenario on digital-corpora [13].

```
# UTF-8 Byte Order Marker; see http://unicode.org/faq/utf_bom.html
# BULK_EXTRACTOR-Version: 1.3.1 ($Rev: 10844 $)
# Feature-Recorder: windirs
# Filename: /data/thesis_data/nps-2008-jean.E01
# Feature-File-Version: 1.1
49997312        GRAM_FIL.ES%        <fileobject src='fat'><atime>2012-10-31T00:00:00</atime><attrib>0</attrib><ctime>2018-02-01T08:26:15</ctime>
50231296        e_sequen.ce'        <fileobject src='fat'><atime>2014-10-12T00:00:00</atime><attrib>32</attrib><ctime>2014-10-19T00:00:00</ctime>
36745728        A0002865.inf        <fileobject src='mft'><atime>2008-05-14T07:04:39Z</atime><attr_flags>0</attr_flags><crtime>2004-08-04T02:09:28Z</crtime>
36746752        A0002113.cpl        <fileobject src='mft'><atime>2008-05-14T05:35:40Z</atime><attr_flags>0</attr_flags><crtime>2004-08-04T04:56:58Z</crtime>
```

Figure 3.2: An excerpt of output from *bulk_extractor* from M57-Jean scenario.

Output

FAT and NTFS filesystems are widely used, making them a high value target for a digital forensics visualization. NTFS filesystems are typically found on desktop and laptop hard drives, it is the filesystem utilized by Microsoft Windows. The FAT filesystem is also of interest to forensic investigators because it is commonly used in mobile and portable media devices as a storage filesystem. As mobile devices are becoming more common in forensic investigations, a method for extracting information, such as timestamps, in an efficient manner is of interest to the community.

The information collected for this visualization from the *windirs* feature file is the modify, access and create (MAC) timestamps for each file and directory entry. The timestamps can be used to ascertain when media was in use assuming the timestamps are accurate. The timestamps are located within the feature file as ISO 8601 formatted date strings contained within XML tags, *mtime*, *atime*, *ctime* and *crtime* within the global scope of a *fileobject* tag. The *ctime* and *crtime* tags present in the *windirs* feature file can indicate different types of timestamps depending upon the *fileobject* type. In the FAT filesystem, *ctime* represents the create time whereas in NTFS, *ctime* represents meta data change time. The *crtime* tag is utilized when parsing NTFS MFT *fileobject* tags in the *windirs* feature file for creation date. The ISO 8601 formatted date strings have different time resolution depending on *fileobject* type. FAT filesystems create timestamps are accurate to within 10 milliseconds, write timestamps to within two seconds and access timestamps to within a day. The NTFS MFT records all timestamps as 64 bit values with a time resolution of 100-nanoseconds [30]. The *fileobject* typically includes other information that is disregarded in this visualization.

The XML output generated by *bulk_extractor* is parsed one line at a time as the feature file is examined from top to bottom. When malformed XML is encountered due to un-escaped character sequences or a tag mismatch the timestamp is disregarded, but the error is printed to the console in case it needs to be referenced for correction. This was useful in correcting an implementation bug in *bulk_extractor*. Each line in the *windirs* feature file contains a well-formed XML document which is parsed individually from the other lines in the document. If one line contains errors it does not effect the parsing of the rest of the document. Parsing of the document in this manner can be time consuming as the XML parser must be reset for each line.

3.4 Visualization

This section describes the graphics technology used for generating the disk activity timeline visualization.

3.4.1 tcpflow

The network summary visualizations in tcpflow, developed by Mike Schick and Dr. Simson Garfinkel, heavily influenced the visualizations that are generated for the disk activity visualization [20]. The code utilized to generate the time histogram was a modified version of tcpflow's packet histogram visualization. Tcpflow uses the cairo library [31] to layout and generate the visualizations for packet analysis in PDF format. The open-source cairo library formats and renders the PDF formatted document. Tcpflow allows for generation of network summary visualizations on a pcap or a live network trace. The graphics presented give the examiner an overview of the network traffic connections, a histogram of types and port numbers in multiple graphs displayed on a single page.

3.4.2 Time Histogram

The time histogram in tcpflow allows for the visualization of events marked with timestamps to be represented by frequency. The frequency counts are contained within time ranges, or buckets, that have a start and end time. As each timestamp is received from the XML parser the type is recorded and processed through the histogram with multiple bucket sizes forming the best fit time ranges. Each bucket represents a snapshot of the frequency of activity during the time period. The buckets are colored in the visualization based upon the type of activity performed during the time period.

The timestamps are placed in sets of buckets that represent common uniform subdivisions of the entire timeline. The first timestamp starts the time range for the first bucket of each set. Subsequent timestamps are tested for overflow of the bucket set time range. If the bucket set is too small to contain the timestamp then the bucket set is discarded and a less granular set is chosen. This allows multiple time resolutions to be computed in a single pass.

The timescale of the tcp histogram objects can represent seconds, minutes, hours, days or years. The flexibility of the histogram objects in tcpflow allowed for minimal modification to the time histogram objects displayed in the disk activity visualization.

During development we determined that it would be useful to display two histograms, one

generated over the entire time range of the timestamps from the *windirs* feature file, the other generated based upon the highest activity bar from the first histogram. The axis in tcpflow are well suited to fast network based packet timeframes, but required modification for the larger timeframes typical of hard drive media. The X axis, time, of each histogram was modified to display at most twenty ticks of time indication. The sparse population of time indicators on the histogram is due to the use of the visualization in triage. The implicit assumption made for time indication was that an investigator would not be concerned about exact times of events during triage, but rather the time frame in which the most events occured.

3.4.3 Cairo

Cairo is a two dimensional graphics library which supports output in a variety of formats [31]. The initial implementation of the disk activity visualization creates a single page PDF file. PDF was chosen as an output format because it is a widely supported file format that allows for high-resolution vector graphics. PDF allows for output to be viewed on a variety of sizes of monitor or printed on a high-quality printer. The PDF file format allows for wide distribution and presentation of the visualization.

CHAPTER 4:
Implementation

This section describes the implementation and problems encountered while developing the disk activity timeline visualization.

4.1 Overview

Histograms were chosen to represent the *windirs* feature file because they show the trend of disk activity over an arbitrary timescale. The timescale displayed in the histogram can be adjusted to any time window using optional start and end time argument flags. The top histogram allows for an overview of the entire timeline of the media or a subset of the timeline given in the optional argument flags. The bottom histogram represents the most active bucket as well as the preceding and succeeding buckets in the timeline. The two histograms can be utilized to focus on the overall activity of the media as well as explore a detailed view of activity during the most active timeframe. The second histogram is automatically generated based upon bucket sizes and cannot be adjusted in the current implementation.

4.2 XML Parser

The *windirs* feature file consists of XML formatted strings containing information about FAT and NTFS MFT entries found within the input media. Complete XML objects are output by the parsing of FAT and NTFS MFT entries by *bulk_extractor*. In order to properly parse the XML formatted objects the C library expat was employed [32]. The expat library is a lightweight stream-oriented XML parser which contains a series of callback handlers to assist in parsing tags found within XML strings. The *windirs* feature file parser registers handlers for specific time based tags found within the XML strings.

An entry in the *windirs* feature file is a complete XML object, as such the *windirs* xml parser is reinitialized upon completion of each entry. The *windirs* feature file parser examines each tag as it is parsed and extracts the mtime, atime and ctime/crtime tags which correspond to modify, access and creation times of the FAT and NTFS MFT file entries. The timestamps present in the time based tags are ISO 8601 formatted date strings which are then transformed into a UNIX timeval. Once the timestamp value and timestamp type are established they are passed into the summary page object to be distributed to the disk activity timeline for rendering.

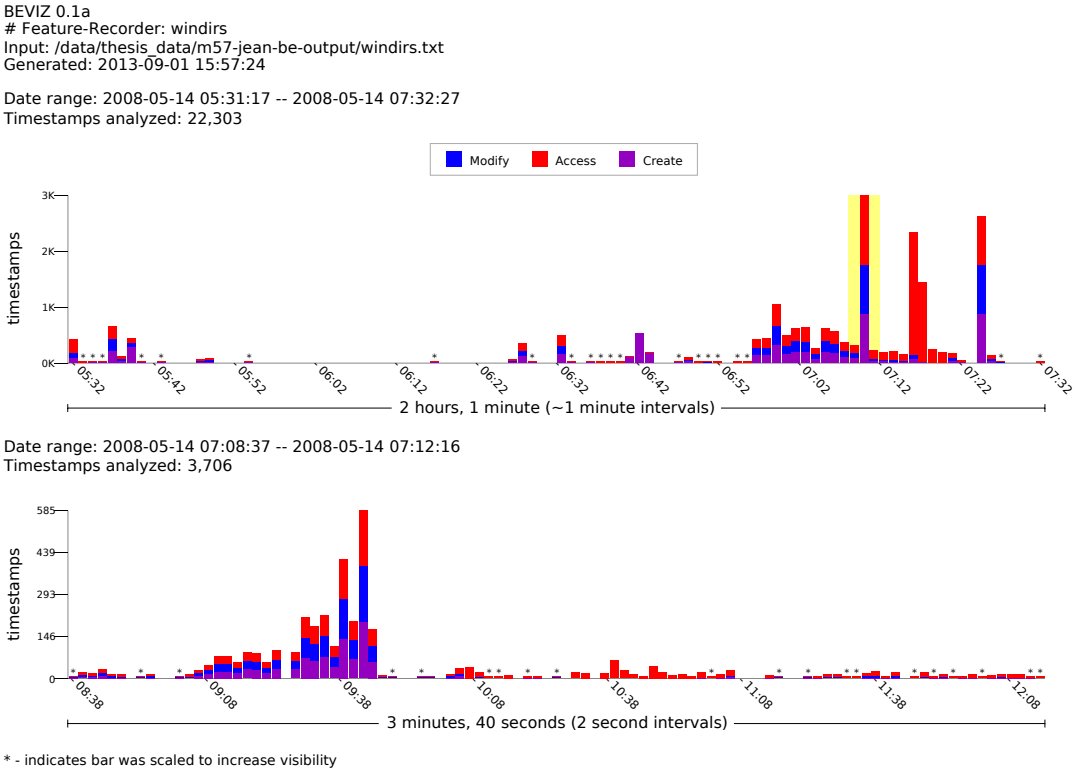


Figure 4.1: Disk activity timeline visualization over a time period of two hours. Generated using start flag of 2008-05-14T05:00:00 and end flag of 2008-05-14T08:00:00 and automatically truncated to the first and last timestamp within the range.

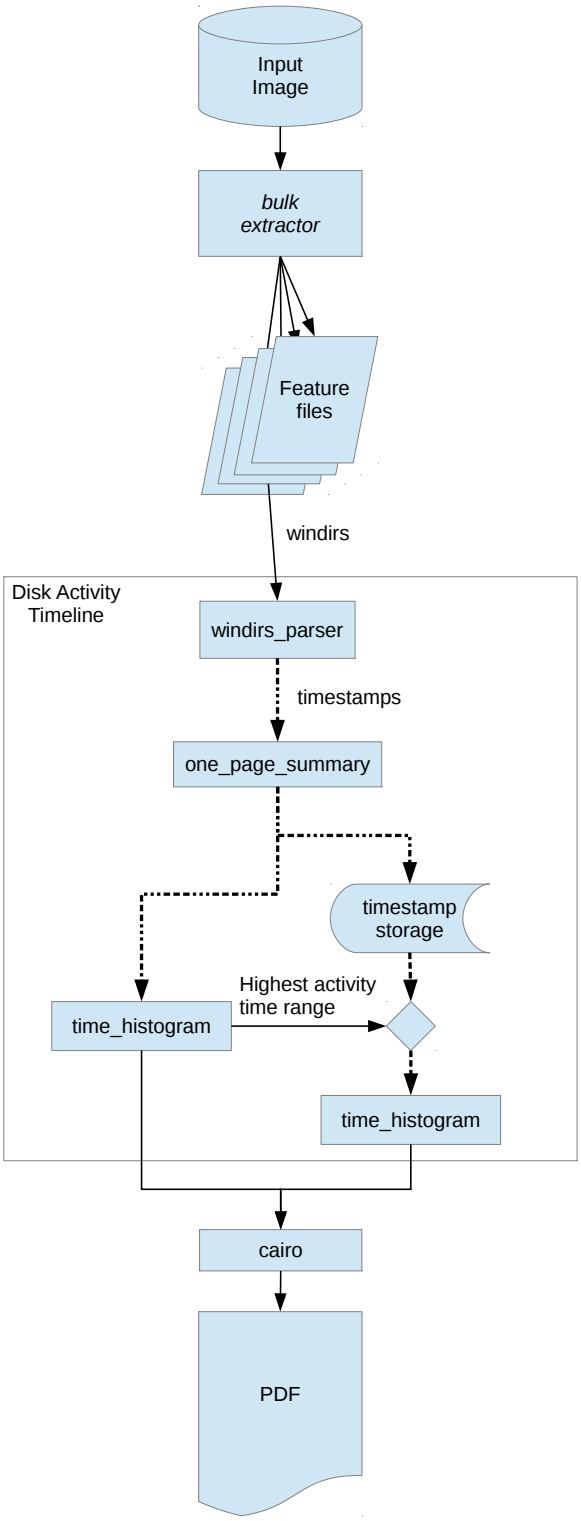


Figure 4.2: Data flow from input disk image to PDF output for the disk activity timeline.

The XML parser encounters errors when XML is incorrectly formatted with respect to the XML schema [33]. When expat encounters any errors it fails to parse the entire XML object and throws an error which is handled by the parser. Output of the error is printed to the screen by the XML parser so that review can be undertaken to fix the incorrect line.

4.3 Filtering

During the course of building the disk activity timeline and testing against real data we discovered that some timestamps produced by *bulk_extractor* are not possible or unlikely to occur, such as timestamps in the distant past or far future. As the *windirs* feature file is processed, any timestamps that occurs before the UNIX epoch is removed as well as any timestamp that occurs in the future in reference to the current time of execution. The removal of the future timestamps is based upon the assumption that timestamps cannot occur in the future during an investigation. All removals of timestamps from the input source, the *windirs* feature file, are printed to the terminal and can be captured in an output file for further review.

4.4 Sorting

As timestamps are generated by the XML parser they are added to time histogram objects through the summary page object. The time histogram objects contain a static sets of bucket configurations. The set of bucket configurations relies on timestamps being in linear order because each timestamp is checked for overflow of the bucket configuration. An example is a set of buckets of 10 seconds per bucket spanning five minutes. Once the first timestamp is added the start time of the first bucket is the same as the first timestamp. When another timestamp is added, if it six minutes later than the first timestamp then the bucket configuration is considered overflowed and a new bucket configuration with larger buckets over a larger time span is chosen. When the timestamps are not processed in a linear ordering the bucket overflow occurs rapidly causing the resulting bucket set to be too large for representing the actual time range of the input timestamps. As timestamps are not guaranteed to be ordered linearly in a FAT or NTFS filesystem, particularly if data has be deleted and reallocated, we have no assumptions of any ordering of the timestamps. Due to the linear ordering requirement of the static sets of bucket configurations, the parsing of the *windirs* feature file must occur in two steps. The first step accumulates the timestamps from the *windirs* feature file and stores them, and the second step sorts these timestamps into a linear order. The sorting of the timestamps incurs the worst case sort time of $N * log_2(N)$ where N is the number of timestamps in the list.

4.5 Time Histogram Generation

The first time histogram to be generated spans the entire time scale of all of the timestamps received from the XML parser, excluding those filtered with flags and time restraints. While the first histogram is being populated by timestamps the summary page object stores the timestamps in a list for the second, highest activity, histogram. Once the first histogram has been completed, the most active bar is found and the timespan between the preceding and succeeding bar is calculated for the representation of the second histogram. The second histogram is generated in the same manner as the first, but a filter for only those dates within the most active timespan is included.

4.5.1 Layout

The disk activity timeline accounts for total activity of all timestamps at each time interval using stacked bars to represent each type of timestamp. The ordering of the bars in the stack represents common nomenclature in the digital forensics field as creations and modifications are considered more important than accesses and placed on the bottom of the stack. A color scheme for quickly identifying different timestamp types and accounting for a large part of the population that is colorblind is necessary. The colors chosen to present the modification, create and access times allow for stark contrast between creation and access types which are stacked on either side of the modification time.

4.6 Histogram Display

The linear ordering of the timestamps due to the bucket generation algorithm can produce histograms that appear different, but represent the same values in different sized buckets. When timestamps are inserted into the histogram buckets they are indexed based upon the bucket sized, the time span of each bucket, scaling of their timestamps. As each bucket is indexed the timestamp count is incremented based upon type.

The rendering process attempts to generate a simple histogram by reducing the number of bars to less than 100 bars for each histogram. The number of buckets with timestamps may be less than this amount, but the non-sparse size of the overall bucket map may include more than 100 buckets. If the maximum bar count is exceeded the buckets are expanded to reduce the non-sparse size of the bucket map to at most 100 bars. Due to the scaling of time values and the expansion of bucket sizes for rendering, certain rounding errors occur during floating point operations to shift bucket counts. While the overall resulting data are still accurate, the rounding

error causes minor shifts in bucket counts in the disk activity timeline.

Linear scaling of the bars in the disk activity timeline was chosen over logarithmic scaling due to the large disparity in bucket counts. Logarithmic scaling causes a sparsely populated disk activity timeline to appear more active by reducing the height of the highest bar and increasing the height of the smallest bars. The goal of the visualization is to be truthful and using a logarithmic scale reduces the recognizable differences in the bucket counts.

The cumulative distribution function(CDF) that is present in the tcpflow network visualizations was removed in the disk activity timeline. The CDF is a measure of the cumulative number of timestamps at each interval along the histogram. The display of the CDF did not add sufficient information to the visualization for the amount of visual noise introduced.

4.7 Histogram Scaling

Buckets in the disk activity timeline that are difficult to visualize due to their small frequency count, in comparison to the largest frequency count, are artificially scaled to be more visible. The artificial scaling increases the vertical height of the bar in relation to the tallest bar in the histogram. The vertical scaling of minimally sized buckets in the disk activity timeline allows for the display of values that are difficult to view on any screen. The scaled buckets are denoted with an asterisk character floating above the bar and explained with a footnote at the bottom of the disk activity visualization. One of the goals of this visualization is to be portable across many viewing platforms and as such, scaling minimal buckets with denotation makes the buckets viewable and understandable.

4.8 Time Axis Demarcation

The disk activity timeline requires timestamps along the X axis of the graphic to give the viewer context as to when events occurred along the histogram. Selecting X axis timestamps to display involves finding a sparse enough timescale for each tick so that labels do not overlap and ensuring the X axis is populated enough to quickly ascertain when a series of actions (MAC timestamps) occurred. The scaling of labels in the X axis are in intervals that are recognizable and hold meaning for forensic investigators. Time scaling is dependent upon the data currently under examination, as an example, an investigator may be reviewing a drive linked to financial fraud in which quarterly activity would be useful in judging relevance of the media to an ongoing investigation. The times scales were chosen for familiarity, such as one day divided into

hours and one year divided into months. The goal of the time scale divisions is to show a maximum of 20 timestamps per disk activity timeline in all cases. The timescales are independent of bar boundaries and occur beside or below a bar on the histogram. The time values along the X-axis represent the two major time components in a timestamp. An example time scale spanning multiple years can have X-axis ticks delimited by Year, a two digit value and Month, a three character abreviation. Smaller time scales will truncate the more granular components of the timestamp for readability.

The X axis in the two disk activity timelines in Figure 4.1 is a representation of two time scales. The first time scale spans two hours one minute and contains X ticks every 10 minutes, making 13 ticks across the time axis. The second time scale spans three minutes 40 seconds and contains X ticks every 30 seconds, making eight ticks across the time axis. Both of the time axis demarcations show recognizable time scales that allow a viewer to ascertain when a given bar of activity occurred without the need for time demarcations on each bar.

4.9 Outliers

When the disk activity timeline is generated without the time scale scope argument flags, the timestamps that are not part of general usage stand out as outliers among the histogram. The outliers can represent the possibility of tampering of the timestamps, unusual activity that may be worth further investigation or a bug in the program that generates them. Further analysis using traditional forensic media analysis methods for identifying tampered timestamps is required to prove tampering.

The visualization currently supports coarse outlier detection and removal. The outlier removal attempts to remove the majority of the erroneous outliers. A secondary method to narrow the disk activity timeline in scope is to employ optional argument flags that specify a start and end time that the histogram should contain.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Validation

The section presents a description of the procedures used to validate the usage of the disk activity timeline visualization.

5.1 Methodology

Validation of the disk activity timeline was accomplished by comparing results from the commercial digital forensics tool FTK and the disk activity timeline generated from a series of disk images. The disk images were generated for other NPS research projects and education for identification of malicious and illicit material. The difficultly of generating a diverse set of hard drive images with similar tracable "criminal" activity limits the amount of validation testing that can occur of the disk activity timeline.

The development branch of version 1.4.1 of *bulk_extractor* was utilized for extracting *windirs* features so that the most accurate timestamps could be used as input to the disk activity timeline. A trial version of the visualization package provided in version 4.02 of FTK was employed as a comparison visualization against the disk activity timeline. The Forensic Toolkit includes a visualization feature with which the files on a disk image can be presented as a histogram. FTK was run against the same NPS generated disk images as above and was used to compare and contrast the results obtained from the disk activity timelines generated by *bulk_extractor*. The *bulk_extractor* output may contain directory entries that are missing from FTK due to the bulk scanning of the disk image. The extra directory entries from *bulk_extractor* are included in the disk activity visualization. The FTK visualizations are based upon an entire filesystem analysis which parses each timestamp from every file found on the disk image. The delay between visualization generation and disk image ingress makes these visualizations unsuitable for triage purposes.

The visualizations presented by FTK and the disk activity timeline display timestamps differently. The FTK visualization chooses to split the modify, access and create timestamps into separate timelines which are displayed in separate graphs. The disk activity visualization combines the modify access and create timestamps into a single graphic and color codes the different timestamp types. To compare the visualizations, the FTK graphics for each timestamp type must be considered to ascertain the highest points of disk activity.

The FTK visualization, by default, displays the entire timeframe for which timestamps are found. In an attempt to highlight the time range for comparison, the window slider was aligned with the date of the start flag in the disk activity timeline. The FTK visualizations do not output in PDF format, as such screen shots were acquired from the FTK application.

Autopsy also includes a visualization feature that displays a time histogram with evidence timestamps. The visualization feature of Autopsy is currently marked beta and exhibited bugs and errors when run against the NPS generated disk images and was not used to validate the disk activity timeline.

5.2 Procedure

The test cases were first run with *bulk_extractor* to find and process the *windirs* features. The *windirs* scanner was the only active feature file scanner during the processing of the test cases. After *bulk_extractor* completed parsing out the FAT and NTFS filesystem metadata into a feature file the time histogram application was run with the feature file as an input. The disk activity timeline was intentionally restricted to a start time of January 1st, 2000 00:00:00 to limit the extraneous timestamps occurring at the MS-DOS epoch (January 1st, 1980 00:00:00). As each disk activity timeline is generated the console output is reviewed for errors or anomalous results that were discarded during processing.

Each disk image was processed by *bulk_extractor* with only the *windirs* feature filter enabled. Once the processing by *bulk_extractor* was complete the output, *windirs* feature file was passed to the disk activity timeline program to generate a PDF output of the disk activity. Once the processing of the *bulk_extractor* feature file was complete the image was copied, via winscp with md5sum verification, to a Windows workstation with FTK installed. The image was then added to a new case in FTK and processed as new evidence. Once processing was completed, all of the evidence files were selected and the visualization function was used to generate a time histogram. The Forensic Toolkit chose to display the timestamp visualization types modify, access and create as separate timelines.

5.2.1 Expected Results

Each test case represented should show a strong correlation to the MAC timestamps produced by The Forensic Toolkit and the narrative. The timelines displayed should represent the most active timestamps as verified by a full filesystem timestamp analysis. In the case of foul play

the disk activity timeline should show that timestamps are out place or warn the user that invalid timestamps were detected.

5.3 Test Cases

The test cases performed with the disk activity timeline are generated for use in education and validation testing. Each test case represents a scenario containing a narrative and evidence in digital format. While the test cases are simulated data they allow for a proof of concept examination into the viability of the disk activity timeline visualization. The test cases presented are useful in measuring the functionality and the applicability of the triage disk activity timeline to digital forensics.

The test scenarios were produced for researchers at the Naval Postgraduate School by summer interns tasked in generating scenarios for education and research. The scenarios attempt to represent simple cases of information hiding surrounding illicit transactions.

5.4 Scenarios

The scenarios are accessible through the digitalcorpora website. Below is a table summarizing information about each scenario and a description of each scenario.

5.4.1 Weapons Scenario One

The scenario involves suspected terrorist weapons purchases. The image was acquired during a raid where the suspect may have tried to hide evidence.

Filename	nps-2011-scenario1.E01
MD5 sum	69c3151fff7e95984e36994a45cd6d60
Image size	37055432923 bytes
Reported file system	NTFS (one partition)
Number of files	127,610(*bulk_extractor*), 136,969(FTK)

Table 5.1: Summary information for the weapons scenario one disk image.

Result

The weapon scenario one *bulk_extractor* disk activity timeline represents on going activity with six peaks of activity. The six peaks have similar quantities of modify, access and create timestamps associated with them. The tallest peak appears to occur around December 2010, but the bucket size is three months so that activity may occur anywhere in that timespan. The second

histogram expands on the tallest peak and its two neighbors and shows that the highest amount of activity occurs around mid-April 2011.

The weapon scenario one Forensic Toolkit visualization is composed of three graphics, one for each timestamp type. The Modified and Created timestamp timelines are similar and show peaks in the first quarter of 2008 and 2011. The Last Accessed timestamp timeline shows a peak in the first quarter of 2011 and smaller peaks towards mid-year 2011.

Neither of the graphics correctly placed peaks around the time at which the illicit files were deleted. A full search of the filesystem reveals orphaned files from deletion around a specific timeframe which was not indicated by either graphic. The omission of the peak on both graphics is possibly due to the larger amount of file timestamps created by operating system files.

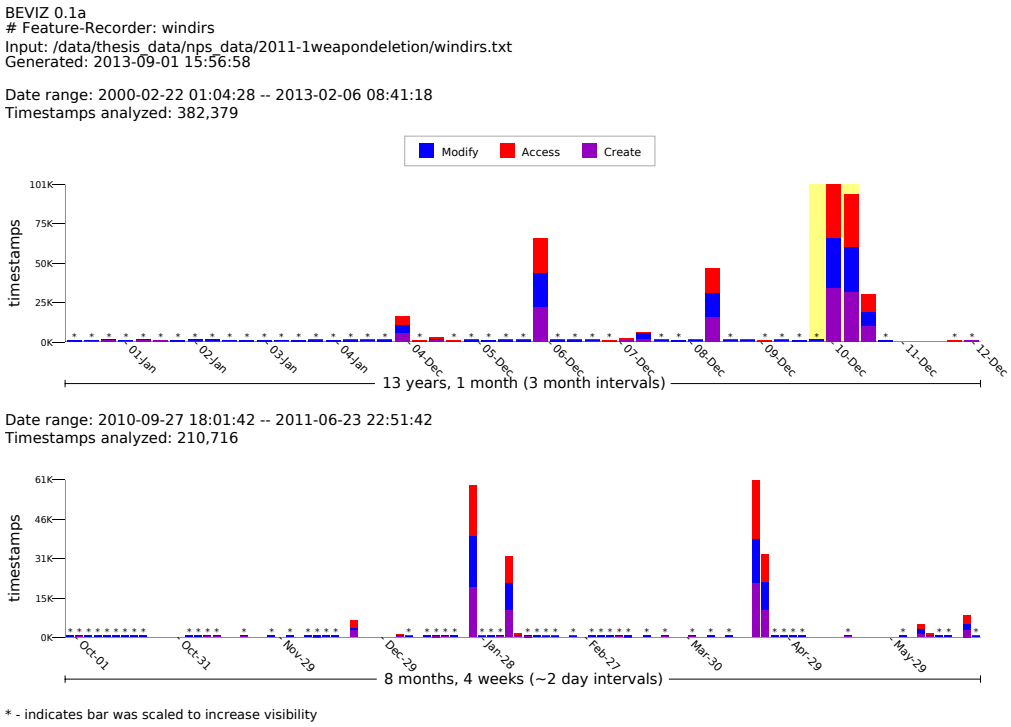


Figure 5.1: Time histogram generated on the drive accompanying the weapons scenario one from *bulk_extractor* output.

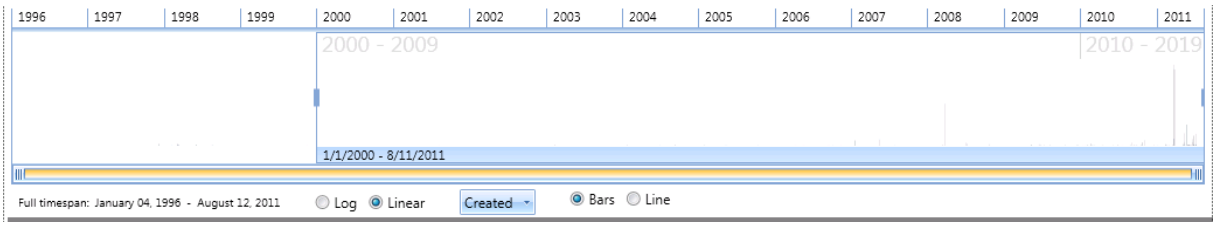


Figure 5.2: Create timestamp histogram generated on the drive accompanying the weapons scenario one from FTK output.

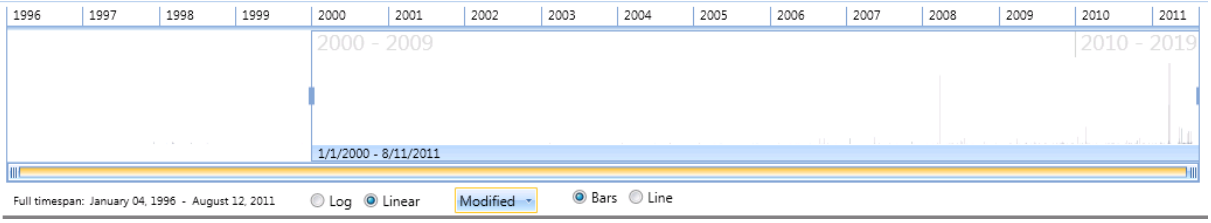


Figure 5.3: Modify timestamp histogram generated on the drive accompanying the weapons scenario one from FTK output.

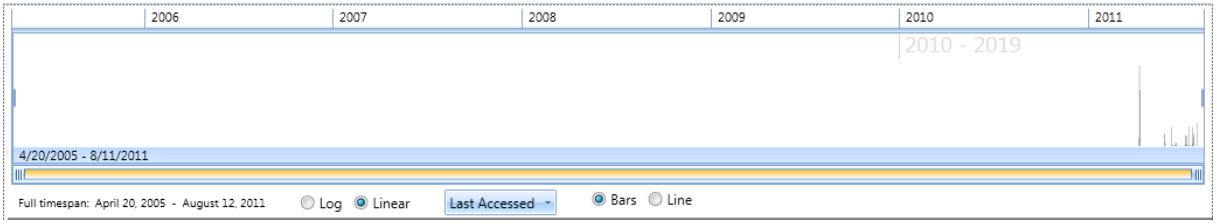


Figure 5.4: Last Accessed timestamp histogram generated on the drive accompanying the weapons scenario one from FTK output.

5.4.2 Weapons Scenario Two

This scenario involves suspected weapons deal between two parties. The image acquired is from one of the parties suspected of transferring information in regards to weapons.

Filename	nps-2011-scenareo2.E01
MD5 sum	23f93eaa648eec7c0683d63dd30b578c
Image size	21427066832 bytes
Reported file system	ext4 (two partitions) and swap (one partition)
Number of files	58,687(*bulk_extractor*, 787,936(FTK)

Table 5.2: Summary information for the weapons scenarion two disk images.

Result

The weapon scenario two *bulk_extractor* disk activity timeline shows sparse activity throughout the timeline with a single peak around December 2009. The single peak along with the two buckets on either side are expanded in the second histogram and appears to show a single day consisting of most of the timestamps found occurring on December 26th 2010.

The Forensic Toolkit visualizations of weapon scenario two show different peaks depending on timestamp type. The Created timestamp timeline shows three peaks of activity in the fourth quarter of 2009 and the first and second to third quarter of 2010. The Modified timestamp timeline shows five peaks of activity in the first quarter of 2009, the second to third quarter of 2009, the last quarter of 2009 and the first and second quarter of 2010. The Last Accessed timestamp timeline shows seven peaks of activity, one in the fourth quarter of 2005, two in 2009, three in 2010 and a final peak in the third to fourth quarter of 2011.

The Forensic Toolkit Last Accessed timestamp timeline includes a peak of activity that encompasses the illegal activity of this scenario. The filesystem for this disk image does not represent the target for the *windirs* feature file extraction which causes the disk activity histogram to omit most file system activity.

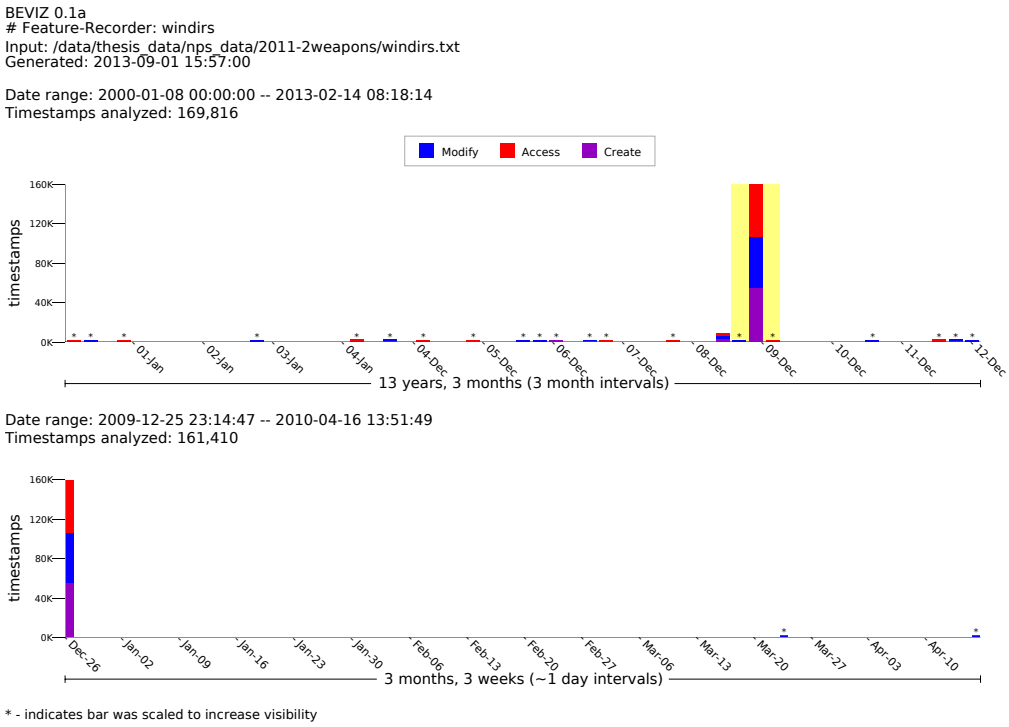


Figure 5.5: Time histogram generated on the drive accompanying the weapons scenario two from *bulk_extractor* output.

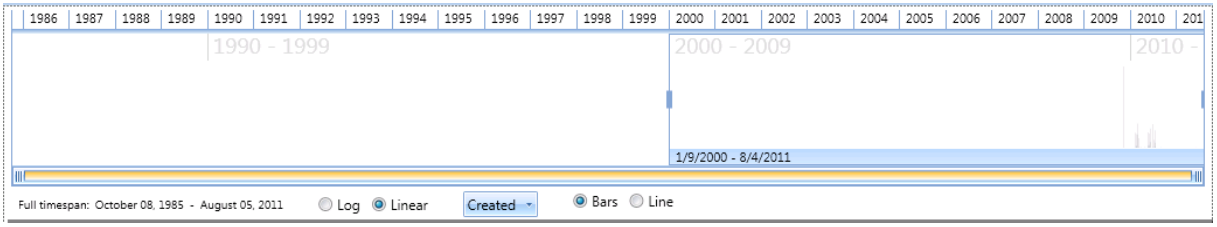


Figure 5.6: Create timestamp histogram generated on the drive accompanying the weapons scenario two from FTK output.

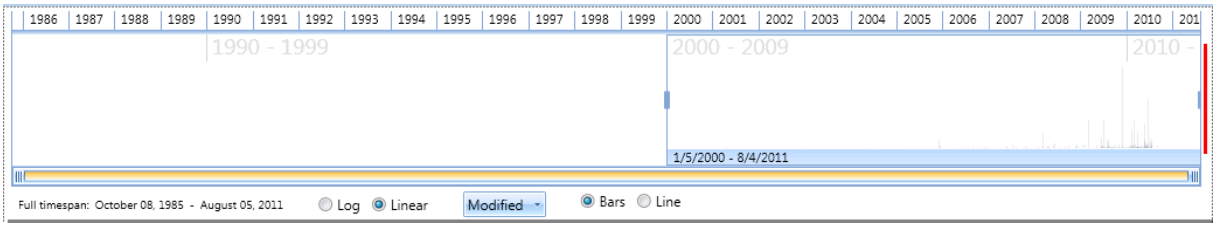


Figure 5.7: Modify timestamp histogram generated on the drive accompanying the weapons scenario two from FTK output.

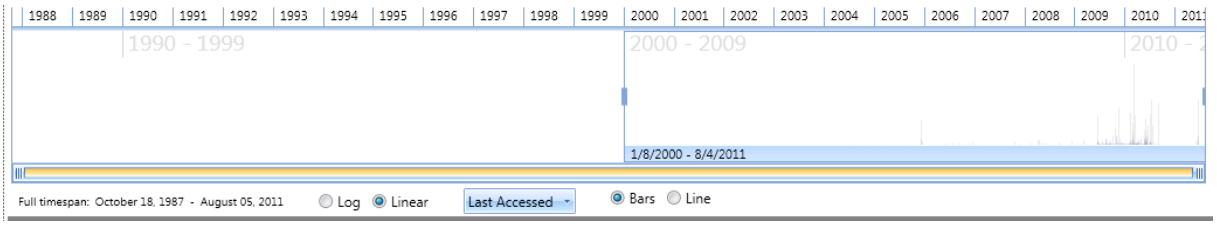


Figure 5.8: Last Accessed timestamp histogram generated on the drive accompanying the weapons scenario two from FTK output.

5.4.3 Drug Traffic Scenario

This scenario involves suspected distribution and procurement of illegal narcotics. The image was acquired without the suspects knowledge.

Filename	nps-2011-scenario4.E01
MD5 sum	390434dfcd182e3c57842a69d45945fe
Image size	19466139106 bytes
Reported file system	NTFS (two partitions)
Number of files	543,518(*bulk_extractor*), 300,921(FTK)

Table 5.3: Summary information for the drug traffic scenario disk image.

Result

During the parsing of the drug traffic scenario *windirs* feature file errors indicating non-well formed xml were output for 25 entries. The errors were not investigated given that the total count of timestamps in the *windirs* feature file was 1,616,804.

The drug traffic scenario *bulk_extractor* disk activity timeline shows sparse activity throughout the disk activity timeline with three distinct peaks occuring around June 2009, December 2010 and June 2010. The largest peak occurs around December 2010 with a three month bucket time span. The second histogram also shows consistent background activity with a single peak around January 20th 2010. The peak is evenly distributed between modify, access and create timestamps.

The Forensic Toolkit visualizations of the drug traffic scenario all show a distinct peak in the middle of the year of 2009. The Last Accessed timestamp timeline also contains a peak for the first quarter of 2011. The Created timestamp timeline includes no other distinct peaks of activity. The Modified timestamp timeline contains an activity peak in the first quarter of 2011.

Neither of the graphics correctly placed peaks around the time at which the illicit files were created or accessed. A full search of the filesystem reveals communications around a specific timeframe which was not indicated by either graphic. The omission of the peak on both graphics is possibly due to the larger amount of file timestamps created by operating system files.

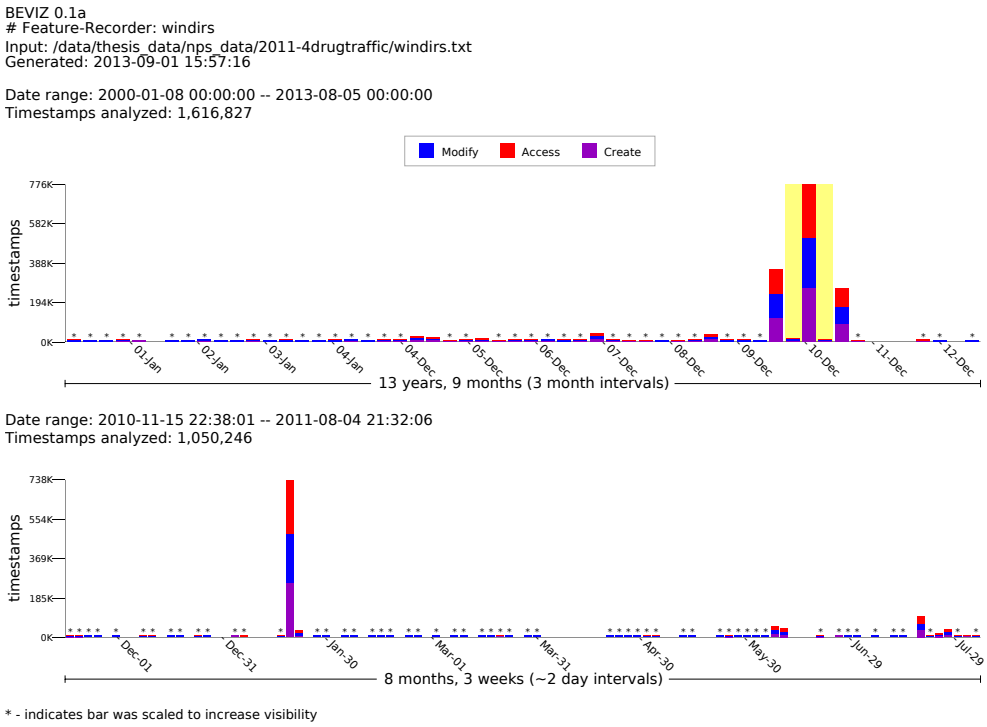


Figure 5.9: Time histogram generated on the drive accompanying the Drug Traffic scenario from *bulk_extractor* output.

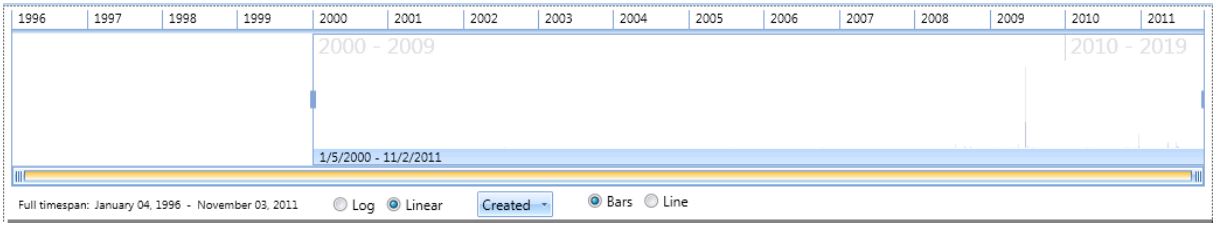


Figure 5.10: Create timestamp histogram generated on the drive accompanying the drug traffic scenario from FTK output.

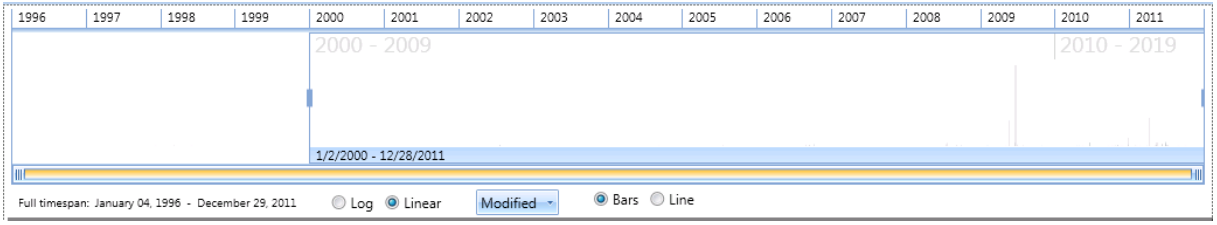


Figure 5.11: Modify timestamp histogram generated on the drive accompanying the drug traffic scenario from FTK output.

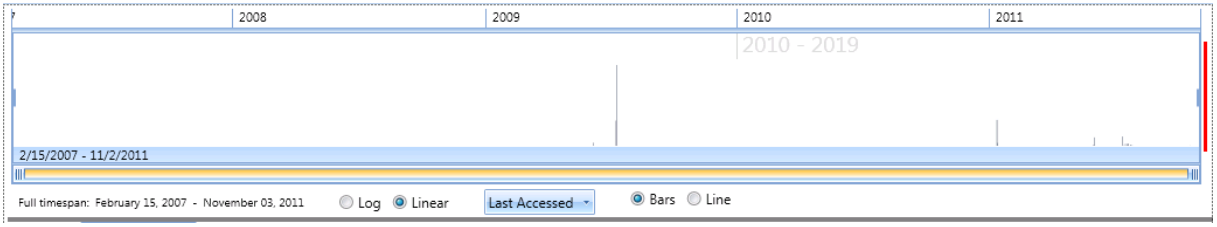


Figure 5.12: Last Accessed timestamp histogram generated on the drive accompanying the drug traffic scenario from FTK output.

Control PC

This disk image is utilized as a control. Typical email and web traffic data was simulated on a computer to show a base case of computer activity.

Filename	nps-2011-scenario5.E01
MD5 sum	3dfc0add50685a28068b4a5ea1994665
Image size	21715396892 bytes
Reported file system	NTFS (one partition)
Number of files	99,754(*bulk_extractor*, 131,176(FTK)

Table 5.4: Summary information for the control pc disk image.

Result

The control PC *bulk_extractor* disk activity timeline show sparse background activity throughout the disk activity timeline with four distinct peaks, three around the last three quarters of 2004. The largest peak occurs in the second quarter of 2010 and is significantly larger than the cluster in 2004. The largest peak and the two buckets on adjoining sides are expanded in the second histogram showing a peak around the 20th of April 2011.

The Modified and Created Forensic Toolkit visualization of the control pc scenario both include a peak in the first quarter of 2008 and a peak in the first to second quarter of 2011. The Last Accessed timestamp timeline include the peak in the first to second quarter of 2011 and a smaller cluster in the third to fourth quarter of 2011.

Both of the graphic contain peaks in the time frame of reported activity on the control pc. The large amount of web and email traffic generated for this disk image was distinguishable from the operating system files.

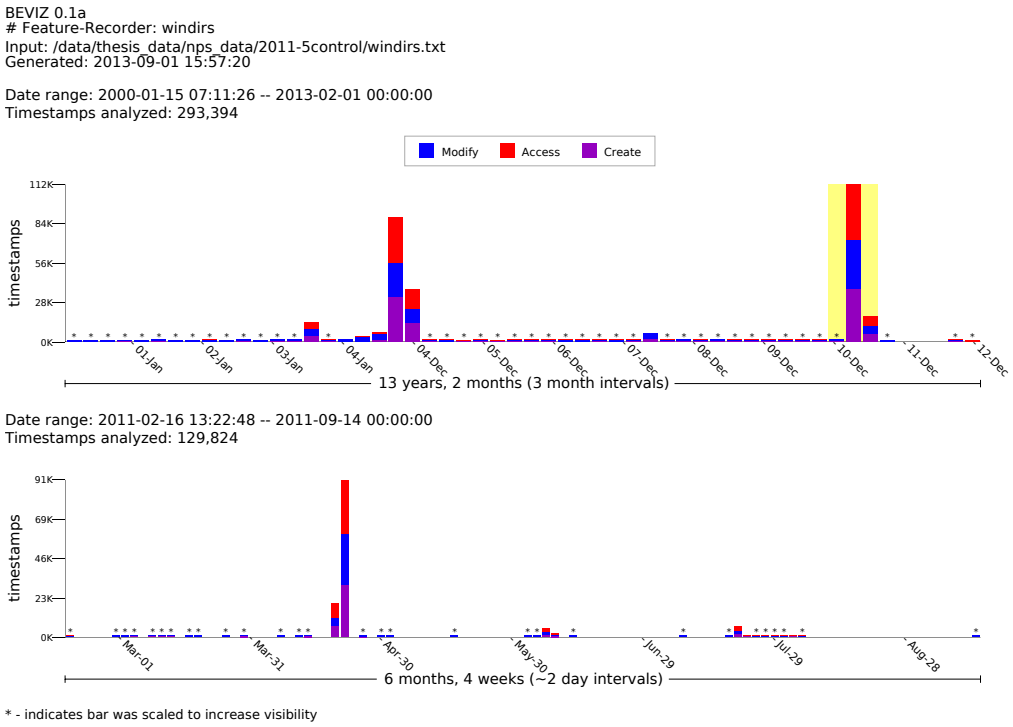


Figure 5.13: Time histogram generated on the drive accompanying the Control scenario from *bulk_extractor* output.

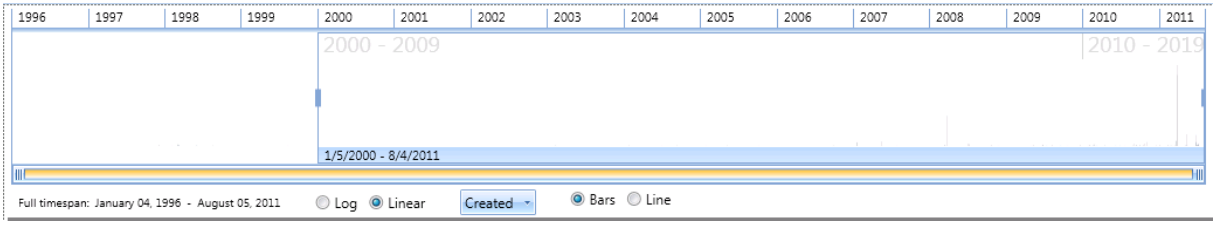


Figure 5.14: Create timestamp histogram generated on the drive accompanying the control pc scenario from FTK output.

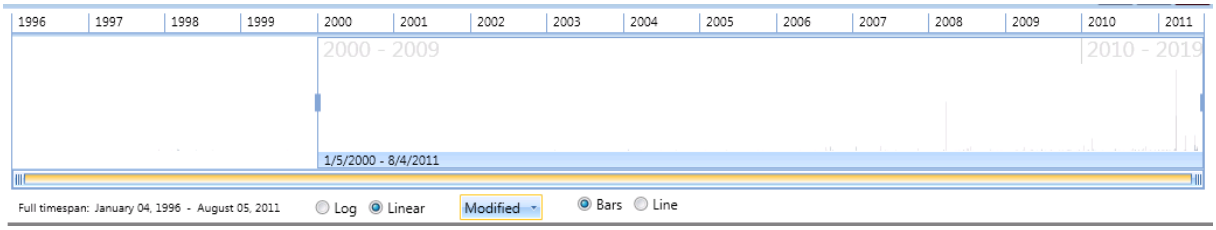


Figure 5.15: Modify timestamp histogram generated on the drive accompanying the control pc scenario from FTK output.

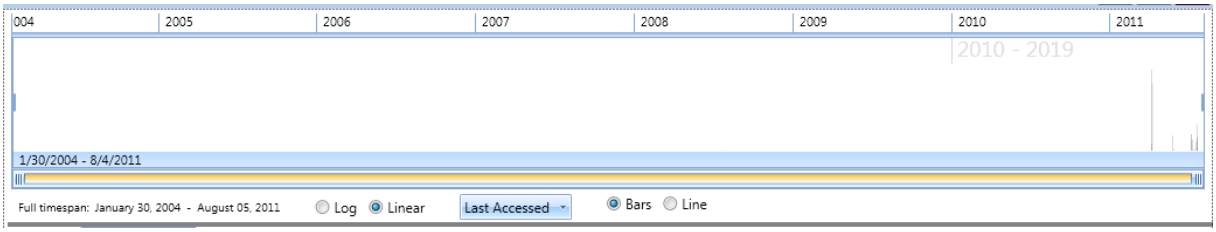


Figure 5.16: Last Accessed timestamp histogram generated on the drive accompanying the control pc scenario from FTK output.

5.5 Post-Analysis

The results from the comparison of the disk activity timeline to the Forensic Toolkit visualization show that peaks can be used to indicate times of high disk activity from filesystem timestamps. The peaks may or may not contain the time frames in which illegal activity occurred, but can assist an investigator in analysis of high throughput activities. The disk activity timeline show if a drive has been active sporadically with distinct peaks of high activity as well as if a drive has been in consistent usage indicated by evenly tall bars. The second histogram gives a detailed view of the bucket with the highest amount of activity and shows where that activity occurs within the bucket from the top histogram. The disk activity timeline is quickly generated from *bulk_extractor* feature files while the visualizations provided by the Forensic Toolkit require a full ingest of evidence. The speed of generating the disk activity timeline makes it suited for use in triage of media.

The illegal activity on scenarios one through four included a small collection of files reduced the possibility of detection by the disk activity timeline and FTK visualization difficult. The large amount of filesystem timestamp activity due to the operating system causes both visualizations to lose the illegal activity in the noise. The removal of operating system activity may increase the disk activity timeline accuracy. Future work to remove known operating system files from the *windirs* feature file during ingress to the disk activity visualization is explained in chapter six.

CHAPTER 6:
Conclusion

This section describes the results obtained from this thesis and future work to extend the current implementation of the disk activity timeline.

6.1 Goals

The goals of this project were to create a simple, efficient, and truthful visualization of disk activity to be used in triage for digital media exploitation. The current environment in digital forensics requires fast, automated analysis of digital media for numerous goals. This project seeks to expand the ability of the digital forensics community to triage incoming digital media in an efficient manner.

The visualization presented in this thesis is able to generate a disk activity timeline for triage use in less time than the commercial tool FTK. The efficiency of the visualization is due to the speed of *bulk_extractor* and the XML parser used to obtain timestamps from the *windirs* feature file. The accuracy of the visualization is similar to the commercial tool FTK and planned improvements aim to increase the efficiency and accuracy of the disk activity timeline. Detection of user-based activities versus operating system operations requires that users generate large peaks of activity which are displayed by both FTK and the disk activity timeline.

6.2 Metrics

The disk activity timeline visualization displays the FAT and NTFS filesystem timestamps that are extracted by *bulk_extractor*. The visual allows for immediate understanding of the most active time ranges during which files were modified, accessed and created on FAT and NTFS filesystem. The extended capability to narrow the timescale can be used by investigators to narrow the focus of the disk activity timeline for better understanding of a specific time interval.

The use of the PDF format for the disk activity visualization allows for scalable graphics that can be rendered on a multitude of display sizes. The scaling of the visualization can cause minor distortion in the timeline graphic. The overall pattern of activity in the timeline visualization can be recognized in both the overall timescale and the highest activity timeline graphics.

The disk activity timeline application was run against the real data corpus [34] to provide thorough testing of parsing and rendering capabilities. The real data corpus contains hard disk images acquired from secondary markets. The average processing overhead of the XML formatted strings is 98 percent of the overall processing time of the visualization. Listed below are totals and averages compiled from running the disk activity timeline across the real data corpus.

- Number of disk images scanned: 2418
- Number of timestamps parsed: 316552290
- Number of timestamps processed: 316218207
- Average timestamps processed/disk: 130917
- Average timestamps utilitized for graphic/disk: 130776
- Average time to process one timestamp: 0.008 milliseconds
- Average time to render: 108 milliseconds
- Average overall runtime: 1.2 seconds

The disk visualization tool can be utilized for fast automated processing of many *bulk_extractor windirs* feature files.

6.3 Limitations

During the implementation of the disk activity timeline technical limitations were encountered. The limitations are outlined in this section and are to be addressed in future work.

6.3.1 Time

The time histogram codebase from tcpflow is based upon time values of the timeval C struct. These time values are based upon UNIX epoch time starting at January 1st 1970. While these time values are consistent with UNIX filesystems, FAT filesystems utilize a date format relative to the MS-DOS epoch of January 1st 1980 [35]. While the timestamps available from the *windirs* feature file can be translated correctly for times before either the UNIX or MS-DOS epoch, the possibility of encountering these timestamps is rare and can be attributed to corruption, program error or malicious intent.

The current implementation of the disk activity timeline requires that ISO 8601 formatted dates from the *windirs* feature file be converted into timeval C structs. The timeval struct allows for simple comparison of time values, but the time values can lose accuracy during conversion. The timeval struct cannot support leap seconds and fails to support those countries that do not follow

the daylight savings time conversions. The current implementation of the disk activity timeline is based upon timeval structs which are generated by pcap during packet header processing [36].

6.3.2 Outlier Detection

Outlier detection among a data set is known to be a difficult problem. Future work to assist the *windirs* feature file parser would extend the current capabilities in outlier detection to contain algorithms in unsupervised clustering. The timestamps that are compiled from the *windirs* feature file are low-dimensional data and can be clustered using density-based clustering algorithms. Implementation of the DBSCAN [37], OPTICS [38] or DeLi-Clu [39] algorithms would allow for efficient removal of outliers in *O(n log n)* time.

6.3.3 Background Activity Reduction

The reduction of unnecessary timestamps created by system files during installation and updates will assist in the removal of outliers. The reduction can be accomplished with a simple list of files that can be safely ignored during the ingestion of timestamps from the *windirs* feature file. The addition a list of files to ignore by the disk activity timeline could allow for greater focus on user activity instead of system activity.

One list of files commonly used in forensic investigations is produced by the NIST Information Techology Laboratory [40]. The list, referred to as the Reference Data Set, is available for download as a series of disk images containing a compressed archive of a csv formatted text file. The text file contains hashes and names of "known, traceable software applications" which can be used to distinguish user generated content from content available from commercial vendors or online resources.

6.3.4 Data Representation

Currently the XML representation of the *bulk_extrator windirs* feature files requires an XML based parser to analyze and form values based on tags present in the strings. While the representation is flexible, an alternative representation with a strict data representation would allow for faster parsing. Future work should focus on generating high speed parsable output formats so that the delay between data acquisition and triage based analysis can be reduced.

To increase processing speed for feature files that contain millions of records another format for the *bulk_extractor* feature files can be considered. An experiment on runtime of parsing

of feature files in JSON and sqlite would in ascertaining the fastest format for extracting post-analysis data. A first step towards this goal would be to transform the current XML documents into different formats and generate parsers for each.

6.4 Future Work

Time constraints while working on this thesis did not allow for a user study with subject matter experts. Feedback obtained from a user study would allow for fine tuning of the disk activity timeline visualizations to general user needs. One approach for a user survey would include soliciting opinions from a web survey from industry persons who would test the tool.

The disk activity timeline, as written, allows for organization and display of any time series data. The current implementation of the disk activity visualization is only concerned with FAT and NTFS filesystem entries and only accounts for timestamps relevant to modify, access and create operations. One proposed addition is to add other timestamps, such as meta-data change time which is present in NTFS Master File Table records. The disk activity timeline can also be used in many context outside the FAT and NTFS filesystem entries. Any feature that *bulk_extractor* can determine valid timestamps from is a candidate for parsing and generating a time histogram for display on a summary page. A logical extension to this work is to determine other feature files that contain timestamped entries that are useful for digital forensic investigators during triage.

To expand the capabilities of the disk activity timeline the processing of multiple folders of *bulk_extractor* output in parallel can be added with optional argument strings. The disk activity timeline can be run in parallel with a thread per *bulk_extractor* output folder to produce multiple disk activity timelines. In addition to the parallel processing of multiple *bulk_extractor* output directories the capability to concatenation multiple directories into a single disk activity timeline visualization should be considered.

6.4.1 Installation

Currently the disk activity visualization is compiled on Linux using a customized makefile. The makefile contains assumptions regarding currently installed libraries and development header files. While this approach has worked under development, a generally useful tool will need to contain an automated configure and build process for UNIX such as GNU autotools [41] and a method for compilation for MS Windows. Future work to encompass the visualization under autotools and mingw [42] will make the tool more useful to a wider user base.

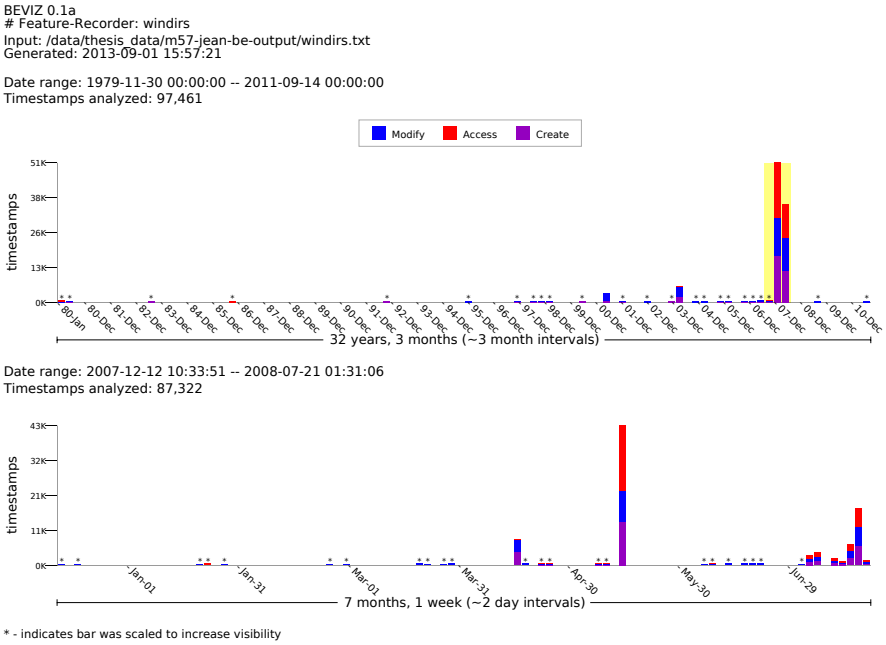
6.4.2 Interactive

Interactive graphics with web-based technologies expand the capability and information available from the visualization presented in this thesis. The addition of interactive components such as hover text and zoom/pan would allow a forensic investigator to extract further details from the visualization as needed. This interactivity would expand the scope of the visualization as a triage tool into the beginning of a post-analysis forensic utility.

THIS PAGE INTENTIONALLY LEFT BLANK

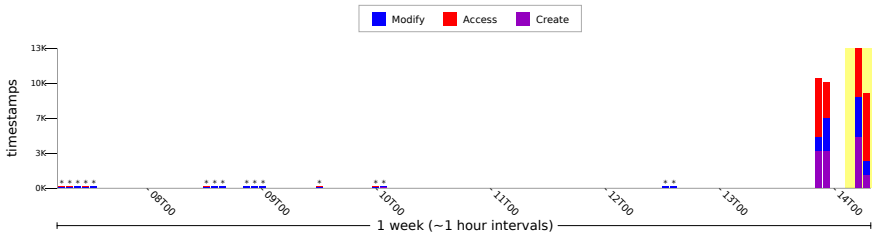
Appendix: M57-Jean Triage Summary

Triage summary visualizations for different time segments generated from *bulk_extractor windirs* feature file from the M57-Jean scenario available on digitalcorpora.org [13].

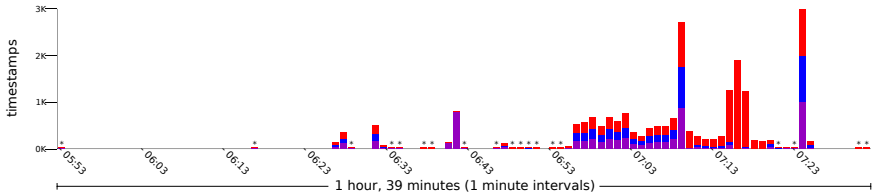


BEVIZ 0.1a
Feature-Recorder: windirs
Input: /data/thesis_data/m57-jean-be-output/windirs.txt
Generated: 2013-09-01 15:57:22

Date range: 2008-05-07 05:04:15 -- 2008-05-14 07:32:27
Timestamps analyzed: 42,889



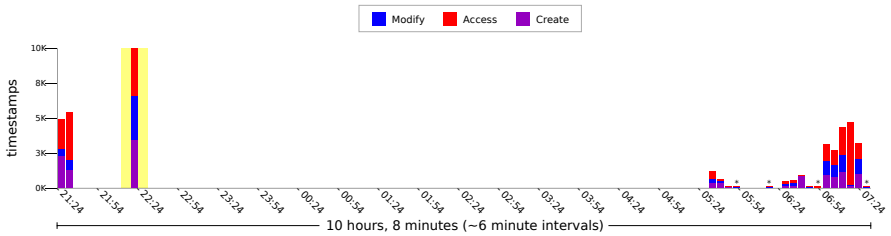
Date range: 2008-05-14 05:52:37 -- 2008-05-14 07:32:27
Timestamps analyzed: 20,296



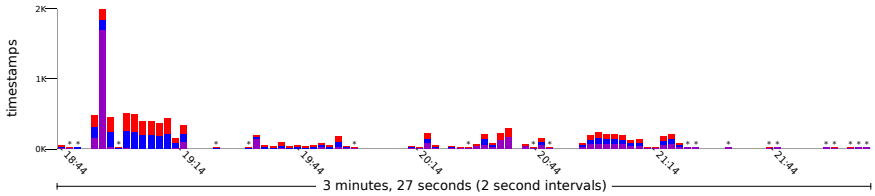
* - indicates bar was scaled to increase visibility

BEVIZ 0.1a
Feature-Recorder: windirs
Input: /data/thesis_data/m57-jean-be-output/windirs.txt
Generated: 2013-09-01 15:57:23

Date range: 2008-05-13 21:23:41 -- 2008-05-14 07:32:27
Timestamps analyzed: 42,813



Date range: 2008-05-13 22:18:43 -- 2008-05-13 22:22:09
Timestamps analyzed: 10,058



* - indicates bar was scaled to increase visibility

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

[1] United States - Computer Emergency Readiness Team. Computer forensics. [Online]. Available: http://www.us-cert.gov/sites/default/files/publications/forensics.pdf

[2] Regional Computer Forensics Laboratory Program. Regional computer forensics laboratory program annual report fiscal year 2011. Accessed: 2013-07-16. [Online]. Available: http://www.rcfl.gov/downloads/documents/RCFL_Nat_Annual11.pdf

[3] E. Casey, *Digital Evidence and Computer Crime*, 2nd ed. San Diego, CA: Elsevier Academic Press, 2004.

[4] C. Ware, *Information Visualization*, 3rd ed. Waltham, MA: Morgan Kaufmann, 2013.

[5] E. Tufte, *The Visual Display of Quantitative Information*, 2nd ed. Cheshire, CT: Graphics Press LLC, 2011.

[6] J. Steele and N. Iliinsky, *Beautiful Visualization*. Sebastopol, CA: O'Reilly Media, Inc, 2010.

[7] N. Yau, *Visualize This*. Indianapolis, IN: Wiley Publishing Inc, 2011.

[8] J. Olsson and M. Boldt, "Computer forensic timeline visualization tool," *Digit. Investig.*, vol. 6, pp. S78–S87, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2009.06.008

[9] Digital Forensics Research Workshop. Dfrws2011 forensic challenge. Accessed: 2013-08-31. [Online]. Available: http://www.dfrws.org/2011/challenge/DFRWS2011_Forensic_Challenge-exported2.pdf

[10] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Digital Investigation*, vol. 9, Supplement, pp. S69 – S79, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S174228761200031X

[11] S. Teerlink and R. F. Erbacher, "Foundations for visual forensic analysis," in *Information Assurance Workshop, 2006 IEEE*, West Point, NY, 2006, pp. 192–199.

[12] Google. Visualization: Treemap. Accessed: 2013-09-01. [Online]. Available: https://developers.google.com/chart/interactive/docs/gallery/treemap

[13] Naval Postgraduate School DEEP Lab. M57-jean. Accessed: 2013-05-05. [Online]. Available: http://digitalcorpora.org/corpora/scenarios/m57-jean

[14] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1–3, pp. 1 – 6, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231298000307

[15] B. Fei et al., "Exploring forensic data with self-organizing maps," in *Advances in Digital Forensics*, ser. IFIP — The International Federation for Information Processing, M. Pollitt and S. Shenoi, Eds. New York, NY: Springer US, 2005, vol. 194, pp. 113–123. [Online]. Available: http://dx.doi.org/10.1007/0-387-31163-7_10

[16] E. J. Palomo et al., "2012 special issue: Application of growing hierarchical som for visualisation of network forensics traffic data," *Neural Netw.*, vol. 32, pp. 275–284, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2012.02.021

[17] K. Lakkaraju et al., "Nvisionip: netflow visualizations of system state for security situational awareness," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. New York, NY: ACM, 2004, pp. 65–72. [Online]. Available: http://doi.acm.org/10.1145/1029208.1029219

[18] QoSient, LLC. argus. Accessed: 2013-07-16. [Online]. Available: http://qosient.com/argus/index.shtml

[19] G. J. Conti, "Countering network level denial of information attacks using information visualization," Ph.D. dissertation, Georgia Institute of Technology, 2006.

[20] S. Garfinkel and S. Michael, "Network forensics with tcpflow," Naval Postgraduate School, Monterey, CA, Tech. Rep. NPS-CS-13-003, 2013, accessed: 2013-05-12.

[21] M. K. Rogers et al., "Computer forensics field triage process model," in *Conference on digital forensics, security and law*, vol. 32, Maidens, VA, 2006.

[22] R. Walls et al., "Forensic triage for mobile phones with dec0de," in *USENIX Security Symposium*. San Francisco, CA: IEEE, 2011.

[23] S. L. Garfinkel, "Digital media triage with bulk data analysis and bulk_extractor," *Computers & Security*, vol. 32, pp. 56 – 72, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404812001472

[24] G. Roussas, "Visualization of client-side web browsing and email activity," Master's thesis, Naval Postgraduate School, Monterey, CA, 2009.

[25] P. Farrell, "A framework for automated digital forensic reporting," Master's thesis, Naval Postgraduate School, Monterey, CA, 2009.

[26] G. Osborne and B. Turnbull, "Enhancing computer forensics investigation through visualisation and data exploitation," in *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, Los Alamitos, CA, 2009, pp. 1012–1017.

[27] AccessData. Forensic toolkit. Accessed: 2013-08-31. [Online]. Available: http://www.accessdata.com/products/digital-forensics/ftk

[28] B. Carrier. Autopsy. Accessed: 2013-08-31. [Online]. Available: http://www.sleuthkit.org/autopsy/

[29] C. Moch and F. C. Freiling, "Evaluating the forensic image generator generator," in *Digital Forensics and Cyber Crime*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, P. Gladyshev and M. Rogers, Eds. Heidelberg, Germany: Springer Berlin Heidelberg, 2012, vol. 88, pp. 238–252. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35515-8_20

[30] Microsoft Developer Network. File times. Accessed: 2013-07-14. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/ms724290(v=vs.85).aspx

[31] B. Esfahbod and C. Worth. Cairo. Accessed: 2013-08-25. [Online]. Available: http://www.cairographics.org

[32] J. Clark. The expat xml parser. Accessed: 2013-05-12. [Online]. Available: http://expat.sourceforge.net/

[33] W3C XML Schema Working Group. Extensible markup language. Accessed: 2013-06-02. [Online]. Available: http://www.w3.org/TR/REC-xml/

[34] S. Garfinkel et al., "Bringing science to digital forensics with standardized forensic corpora," in *Digital Investigation*, vol. 6, Supplement, 2009, pp. S2 – S11, the Proceedings of the Ninth Annual DFRWS Conference. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287609000346

[35] Microsoft Corporation. Microsoft efi fat32 file system specification. Accessed: 2013-05-12. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/hardware/gg463080.aspx

[36] T. Carstens. Programming with pcap. Accessed: 2013-05-27. [Online]. Available: http://www.tcpdump.org/pcap.htm

[37] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, 1996.

[38] M. Ankerst et al., "Optics: ordering points to identify the clustering structure," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999.

[39] E. Achtert et al., "Deli-clu: Boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking," in *Advances in Knowledge Discovery and Data Mining*. Heidelberg, Germany: Springer Berlin Heidelberg, 2006, vol. 3918, pp. 119–128. [Online]. Available: http://dx.doi.org/10.1007/11731139_16

[40] National Institute of Standards and Technology. National software reference library. Accessed: 2013-08-11. [Online]. Available: http://www.nsrl.nist.gov/

[41] Free Software Foundation. Automake. Accessed: 2013-07-18. [Online]. Available: http://www.gnu.org/software/automake/

[42] C. Peters. Mingw. Accessed: 2013-07-18. [Online]. Available: http://www.mingw.org/

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia
2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California